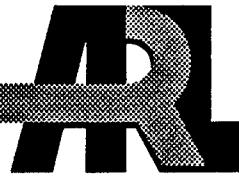


ARMY RESEARCH LABORATORY



Predicting Coalescence of Blast Waves From Sequentially Exploding Ammunition Stacks

John Starkenberg
Kelly J. Benjamin

ARL-TR-645

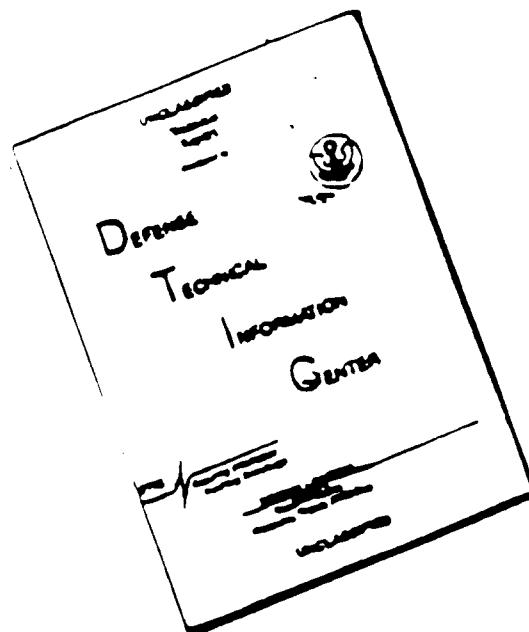
December 1994



19941229 048

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST
QUALITY AVAILABLE. THE COPY
FURNISHED TO DTIC CONTAINED
A SIGNIFICANT NUMBER OF
PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute endorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	December 1994	Final, 1 October 1992-30 June 1994	
4. TITLE AND SUBTITLE		5. FUNDING NUMBERS	
Predicting Coalescence of Blast Waves From Sequentially Exploding Ammunition Stacks		4G031-402-T5	
6. AUTHOR(S)			
John Starkenberg and Kelly J. Benjamin			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER	
U.S. Army Research Laboratory ATTN: AMSRL-WT-TB Aberdeen Proving Ground, MD 21005-5066			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
U.S. Army Research Laboratory ATTN: AMSRL-OP-AP-L Aberdeen Proving Ground, MD 21005-5066		ARL-TR-645	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION / AVAILABILITY STATEMENT		12b. DISTRIBUTION CODE	
Approved for public release; distribution is unlimited.			
13. ABSTRACT (Maximum 200 words)			
<p>The current requirement that quantity-distance computations for air blast limitations be based on the total weight of all mass-detonating explosives at a storage site may be too restrictive. Therefore, we developed a computer program called "BWACO," intended to estimate pertinent aspects of the blast environment associated with sequentially detonating, spatially distributed ammunition stacks. This report explains the assumptions used and documents the evolution of BWACO on the Cray following its initial implementation. Comparison of preliminary results with experimental data obtained by Zaker led to replacement of the standard initially used for the description of blast waves with a new standard based on experimental data. BWACO has been adapted for the personal computer with enhanced graphical representations. Application to a number of problems representative of typical ammunition storage configurations are detailed. The results indicate that regions of significant pressure associated with the coalescence of blast waves from distributed ammunition stacks may be less extensive than corresponding regions associated with regulatory requirements. An advantage associated with the distribution of ammunition into smaller subdivisions was also demonstrated. As currently configured, BWACO provides a means of assessing the blast environment associated with the sequential detonation of an arbitrary arrangement of ammunition stacks. The limitations imposed by the assumptions have not been assessed in realistic configurations.</p>			
14. SUBJECT TERMS		15. NUMBER OF PAGES	
blast, blast waves, blast loads		108	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

ACKNOWLEDGMENT

The work reported herein was supported under the Safeload Program by the Project Manager for Ammunition Logistics, Picatinny Arsenal, NJ. Safeload is ably managed by Mr. Robert A. Rossi.

Accesion For	
NTIS	CRA&I
DTIC	TAB
Unannounced	
Justification _____	
By _____	
Distribution / _____	
Availability Codes	
Dist	Avail and / or Special
A-1	

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENT	iii
LIST OF FIGURES	vii
LIST OF TABLES	xi
1. BACKGROUND	1
2. APPROACH	1
3. BWACO ASSUMPTIONS	2
4. USING BWACO	4
5. EVOLUTION OF THE BWACO MODEL	5
5.1 Review of Experimental Results	5
5.2 Simulation of the Experiments Using the 1-KT Standard	6
5.3 Alternatives to the 1-KT Standard Fits	8
5.4 Simulation of the Experiments Using the Alternative Standard	12
5.5 Effect of Pressure Combination Algorithms	12
5.6 Simulation of the Experiments Using the Alternative Standard and Pressure Combination Algorithm	17
5.7 Results With Multiple Initial Donors	17
6. PERSONAL COMPUTER VERSION GRAPHICS	24
7. SIMULATIONS OF REPRESENTATIVE LARGE-SCALE CONFIGURATIONS ..	29
8. SUMMARY AND CONCLUSIONS	41
9. REFERENCES	45
APPENDIX A: DETAILED DESCRIPTION OF BWACO SOURCE CODE	47
APPENDIX B: BWACO USER'S MANUAL	85
DISTRIBUTION LIST	103

INTENTIONALLY LEFT BLANK.

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Coalescence maps for Zaker experiment simulations using the 1-KT standard	7
2. Comparison of the 1-KT standard with the alternative standard for shock arrival time	9
3. Comparison of the 1-KT standard with the alternative standard for positive phase duration	10
4. Comparison of the 1-KT standard with the Kingery fit with the Kingery and Zaker data for peak overpressure	11
5. Coalescence maps for Zaker experiment simulations using the alternative standard ..	13
6. Coalescence maps showing the effect of scale on coalescence at two different delay times	14
7. Coalescence maps for Zaker experiment simulations with coalescence indications suppressed at low peak overpressure	15
8. Comparison of the effects of the pressure combination algorithm on the region of significant pressure as a function of the number of charges	16
9. Coalescence maps for Zaker experiment simulations using the alternative standard and pressure combination algorithm	18
10. Coalescence maps for Zaker experiment simulations with unequal charges having weight ratios of 1:2	19
11. Coalescence maps for Zaker experiment simulations with unequal charges having weight ratios of 2:1	20
12. Coalescence maps for Zaker experiment simulations with unequal charges having weight ratios of 1:2 and 2:1	21
13. Coalescence maps for Zaker experiment simulations with three equal charges	22
14. Coalescence maps for Zaker configuration simulations with two equal charges and multiple donors	25
15. Coalescence maps for Zaker configuration simulations with two unequal charges and multiple donors	26

<u>Figure</u>		<u>Page</u>
16.	Coalescence maps for Zaker configuration simulations with three equal charges and multiple donors	27
17.	Coalescence and peak overpressure maps from the PC version of BWACO	28
18.	Coalescence maps for large-scale simulations with two 50,000-lb stacks	30
19.	Peak overpressure maps for large-scale simulations with two 50,000-lb stacks	31
20.	Coalescence maps for large-scale simulations with two unequal stacks having a total weight of 100,000 lb	34
21.	Peak overpressure maps for large-scale simulations with two unequal stacks having a total weight of 100,000 lb	35
22.	Coalescence maps for large-scale simulations with four 25,000-lb stacks	36
23.	Peak overpressure maps for large-scale simulations with four 25,000-lb stacks	37
24.	Coalescence maps for large-scale simulations with sixteen 6,250-lb stacks	38
25.	Peak overpressure maps for large-scale simulations with sixteen 6,250-lb stacks	39
26.	Coalescence and peak overpressure maps for a large-scale simulation with six unequal stacks having a total weight of 150,000 lb	40
27.	Coalescence and peak overpressure maps for a large-scale simulation with ten unequal stacks having a total weight of 250,000 lb	42
A-1.	Schematic of BWACO subroutine calling relationships	50
A-2.	Subroutine <i>at62</i> from the waveform module	52
A-3.	Subroutine <i>sckt</i> from the waveform module	53
A-4.	Functions <i>dist</i> , <i>tshk</i> , and <i>dtpp</i> from the waveform module	54
A-5.	Functions <i>peak</i> , <i>reak</i> , and <i>veak</i> from the waveform module	55
A-6.	Subroutine <i>wave</i> from the coalescence module	57
A-7.	Subroutines <i>ssup</i> , <i>lamb</i> , and <i>comb</i> from the coalescence module	58
A-8.	Subroutine <i>coal</i> from the coalescence module	59
A-9.	Subroutine <i>yeff</i> from the coalescence module	61

<u>Figure</u>		<u>Page</u>
A-10.	Subroutine <i>effy</i> from the coalescence module	62
A-11.	Function <i>tdly</i> from the propagation module	63
A-12.	Subroutine <i>arly</i> from the propagation module	64
A-13.	Subroutine <i>prop</i> from the propagation module	66
A-14.	Subroutine <i>inpt</i> from the execution module	67
A-15.	Subroutine <i>sseq</i> and function <i>cent</i> from the execution module	68
A-16.	Subroutine <i>stak</i> from the execution module	69
A-17.	Subroutine <i>blok</i> from the execution module	70
A-18.	Subroutine <i>stns</i> from the execution module	71
A-19.	Subroutine <i>rtim</i> from the execution module	72
A-20.	Subroutine <i>copt</i> from the execution module	74
A-21.	Subroutine <i>rest</i> from the plotting module	75
A-22.	Subroutine <i>spmn</i> from the plotting module	76
A-23.	Subroutine <i>smmn</i> from the interface module	77
A-24.	Subroutine <i>rcfg</i> from the interface module	78
A-25.	Subroutines <i>savn</i> and <i>geto</i> from the interface module	79
A-26.	Subroutines <i>pmen</i> and <i>mmen</i> from the interface module	81
A-27.	Program <i>bwac</i> and subroutines <i>cnfg</i> and <i>exec</i> from the interface module	82
A-28.	Subroutine <i>cnst</i> from the interface module	83
B-1.	Arrangement of four 25,000-lb ammunition stacks	94
B-2.	A coordinate system for this arrangement	94
B-3.	Coalescence map	96
B-4.	Peak overpressure map	96

INTENTIONALLY LEFT BLANK.

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1.	Variation of Blast Hazard Areas with Separation for Two 50,000-lb Stacks	29
2.	Variation of Blast Hazard Areas with Explosive Weight Distribution for Two Stacks Having a Total Weight of 100,000 lb Separated by 150 m	32
3.	Variation of Blast Hazard Areas with Separation for Four 25,000-lb Stacks	32
4.	Variation of Maximum Effective Explosive Weight and Blast Hazard Areas with Separation for Sixteen 6,250-lb Stacks	33

INTENTIONALLY LEFT BLANK.

1. BACKGROUND

In most instances, Army Regulation (AR) 385-64 requires that quantity-distance computations to determine ammunition storage siting limitations be based on the total weight of all mass-detonating explosives at the site. This requirement is made under the assumption that the blast waves emanating from storage subdivisions (ammunition "stacks") as they sequentially detonate will coalesce with one another before their amplitudes become negligible. If it can be shown that barriers between stacks prevent or sufficiently delay propagation of detonation between them, distances may be determined using the explosive weight of each stack individually. The regulation provides a crude criterion, based on initiation delay, for determining when coalescence will occur for the explosion of two closely spaced stacks. This criterion has not been verified for complicated ammunition storage arrangements and may be too restrictive, particularly where significant spatial separation of stacks exists.

Experience with the quantity-distance requirements has shown that they are often difficult to meet and waivers of the regulations have been frequently requested. If the actual coalescence pattern from a complex array of ammunition stacks could be estimated, a more accurate assessment of the hazards produced could be made, and significant reductions in real-estate requirements might result. In principle, this could be accomplished by numerical simulation of the blast environment in the region of interest. However, this approach requires immense computer resources, even for the simplest arrangements, and is beyond the reach of the typical user. Alternatively, approaches based on simplified analyses which have been developed in the nuclear blast community could be applied. Specifically, algorithms found in the DNA Nuclear Blast Standard (1 KT) (Needham and Crepeau 1981), a computer program which describes the blast environment produced by a 1-KT nuclear explosion, and the Low Altitude Multiple Burst (LAMB) model, a computer program which describes the blast environment produced by multiple nuclear explosions of arbitrary yield, served as the starting point for the present approach.

2. APPROACH

We originally intended to adapt these programs to our present purposes. However, direct use of LAMB required the performance of many unnecessary, time-consuming computations. Therefore, we developed a new application called BWACO, intended to estimate pertinent aspects of the blast environment associated with sequentially detonating, spatially distributed ammunition stacks.

In its initial implementation, BWACO made significant use of 1-KT Standard and LAMB algorithms. However, as we investigated the performance of the code, these were gradually abandoned in favor of new algorithms. The main development effort was conducted using ARL's Cray X/MP 48 because graphics software was readily available. However, once a FORTRAN-callable graphics package was obtained, the code was converted for use on IBM Personal Computers (PCs) or compatible machines.

This report explains the assumptions used and documents the evolution of BWACO on the Cray following its initial implementation. Features added in the PC version are illustrated through application to a number of problems representative of typical ammunition storage configurations. A detailed description of the code (including subroutine listings) and the initial versions of the BWACO User's Manual are presented in the appendices.

3. BWACO ASSUMPTIONS

Sequential detonations are assumed to occur as a result of propagation of detonation from one stack to another. The first stack of any such pair to detonate is called the donor and the second is called the acceptor. The first of all the stacks to detonate is referred to as the initial donor. Detonation propagation is assumed to occur at a constant velocity which is a property of the donor. This is consistent with a fragment impact mechanism at least over distances short enough to preclude significant deceleration of the fragments. The velocity provided must be that of the fastest fragment produced by munitions in the stack augmented to account for focussing effects. The fact that no consideration has been given to aerodynamic fragment decelerations or to additional delays associated with the penetration of barriers makes this a worst-case assumption (when sufficiently high fragment velocities have been specified).

Classical blast wave structure is assumed. The blast wave consists of a shock wave and positive overpressure phase followed by a negative overpressure phase (when sufficiently far from the source). The usual scaling laws are assumed to apply and are used to determine required blast wave characteristics with reference to a standard. BWACO requires shock and zero-overpressure arrival times at specified stations. The 1-KT Standard gives the shock and zero-overpressure radii as functions of time based on fits to analytical solutions. These functions may be inverted using Newton's method to obtain the arrival times. We developed alternative fits to experimental data for a somewhat different standard which provides the shock arrival time and positive phase duration as functions of position. These can be used directly to compute shock and zero-overpressure arrival times.

The ground plane is assumed to be perfectly flat and rigid and to effectively double the explosive weight of a stack. The latter is also a worst-case assumption. Waves from multiple sources are assumed to propagate independently from one another, and coalescence is assumed to occur wherever and whenever the shock associated with one blast wave encroaches into the positive overpressure phase associated with another blast wave.

Under these assumptions, regardless of the spatial and temporal separation between the sources, at sufficiently large distances from the charges, coalescence will be predicted. Blast waves from actual detonating stack arrangements do not exhibit this property because they do not propagate independently. Rather, portions of some of them propagate in air processed by their predecessors. An overtaking wave may first encounter the negative overpressure phase of the wave being overtaken causing it to decelerate. The overtaking wave must be somewhat stronger than the wave being overtaken or it will not be able to penetrate the negative overpressure phase and coalescence cannot occur. Once this phase has been penetrated and the overtaking wave has entered the positive phase of the wave being overtaken, coalescence is assured but occurs only after some additional propagation. For these reasons, the BWACO coalescence algorithm can be expected to predict coalescence where, in actuality, it does not occur. A margin of safety is provided by prediction of a larger hazard area than actually exists.

We considered several methods of combining peak overpressures from N coalesced waves. These include simple superposition,

$$\Delta p = \sum_{i=1}^N \Delta p_i,$$

the full LAMB algorithm,

$$\rho = \rho_0 + \sum_{i=1}^N \Delta \rho_i,$$

$$\nabla = \frac{\sum_{i=1}^N \rho_i \nabla_i}{\rho},$$

$$\Delta p = \sum_{i=1}^N \Delta p_i + \frac{\gamma+1}{2} \left(\frac{1}{2} \sum_{i=1}^N \rho_i V_i^2 - \frac{1}{2} \rho V^2 \right),$$

or use of the pressure produced by the total explosive weight of all stacks associated with the coalesced waves combined at their center of charge. This last approach is described in greater detail in section 5.5.

4. USING BWACO

In order to run BWACO, the user supplies a description of the distributed ammunition stacks and, if desired, of the region of interest. The program divides the region into discrete stations. It is recognized that, generally, any stack may act as the initial donor and it is necessary to determine the worst-case loading (highest combined peak overpressure) at each station considering all initial donors. For each initial donor, BWACO determines the timing of detonation of each of the other stacks. It then applies the coalescence criterion and computes both the total equivalent weight of the explosive in the stacks contributing to the blast and the peak overpressure at each station. It retains the values associated with the highest peak overpressure for any initial donor. The results are reported in graphical form.

User specifications are supplied by means of an appropriately formatted input file. Each ammunition stack is defined by specifying its coordinates in a user-defined Cartesian system, its net equivalent explosive weight, a propagation velocity for communication of mass detonation to neighboring stacks, and whether or not the stack may act as an initial donor. Any convenient Cartesian coordinate system may be used with distances specified in meters. The explosive weight for the TNT equivalent, in pounds, of all the explosive in the stack is required, as is the propagation velocity in m/s. Usually, in order to allow determination of the worst-case blast loading, all of the stacks should be allowed to act as initial donors. However, this input option provides the flexibility required to address special problems such as simulating experiments with known donors. The extent of the region of interest is specified using the same coordinate system defined for the stacks. If no region is specified, BWACO selects a region limited to the area in which the total equivalent weight of the explosive in all the stacks produces peak blast overpressures above the allowable value for inhabited buildings (unless similar considerations for the individual stacks dictate a larger region). The program then positions 3,600 stations within the region.

BWACO cycles through all possible initial donors as specified by the user. For each initial donor, using the given propagation velocities, it computes the times at which the other stacks detonate. It applies the coalescence criterion and, if necessary, combines the peak overpressures at each station.

BWACO provides different graphical representations of the results depending on the version used. The coalescence map is a plot showing areas of coalescence within the region of interest and indicating,

in the Cray version, the number of waves which have coalesced or, in the PC version, the total equivalent explosive weight contributing to the blast. The peak overpressure map, available only in the PC version, is a contour plot showing the worst-case peak overpressure from the blast in the region of interest. The contour intervals correspond to the five permissible exposure levels defined in AR 385-64.

5. EVOLUTION OF THE BWACO MODEL

5. 1 Review of Experimental Results. Zaker (1969) reported results of an extensive analytical and experimental study of the coalescence of blast waves produced by pairs of sequentially detonating Composition C-4 charges having a total weight of two pounds. In the analysis, the charges were assumed to be located at the same point, while in the experiments they were separated by 10 inches and a steel barrier to prevent sympathetic detonation. Pressure was measured along a lateral line equidistant from the charges and along an axial line. The analysis was assumed to be applicable to the experiments along the lateral line. The axial values measured depend on the separation between the charges and are affected by the presence of the barrier, while the lateral values are approximately independent of the separation and are predicted by the analysis.

Zaker considered charge weight ratios for successively detonated charges of 1:2, 1:1, and 2:1 and nominal initiation delay times ranging from 0.8 to 5.7 ms corresponding to 0.60 to $4.11 \text{ ms/lb}^{1/3}$ using conventional blast scaling. From this it can be inferred that an equivalent TNT weight of about 2.62 lb was used in the scaling. The blast environment was monitored using pressure gauges on the axis of charge centers and in the lateral direction perpendicular to this axis at the center of charge to a distance of approximately 58 ft ($42 \text{ ft/lb}^{1/3}$). This distance is a little shorter than that at which the overpressure from the combined charges decays to about 0.9 psi. This represents the smallest permissible exposure level (which applies to inhabited buildings) defined in AR 385-64.

Pressure records showing the coalescence process were presented by Zaker. These indicate that substantial propagation, covering most of the field of observation, may occur as coalescence progresses. He observed that the peak overpressures associated with coalesced waves were essentially the same as those produced by single charges of the same total explosive weight.

For two charges of equal mass, Zaker found that a "tendency" to coalescence in the lateral direction occurs for delays of less than 4.3 ms ($3.2 \text{ ms/lb}^{1/3}$). The term "tendency" was not clearly defined but it may be assumed to mean observation of decreasing intervals between wave peaks with increasing distance

from the charge center. This is actually very similar to the BWACO criterion. He observed a tendency to coalescence in the axial direction with all of the delays considered (up to 5.7 ms or $4.1 \text{ ms/lb}^{1/3}$). By comparing lateral and axial data, he estimated that the effect of the charge separation and the barrier on the interval between peaks in the axial direction is equivalent to an additional delay of 1.8 ms ($1.3 \text{ ms/lb}^{1/3}$). Thus, axial coalescence may be assumed to occur for delays less than 6.1 ms ($4.5 \text{ ms/lb}^{1/3}$). This result may include significant effects of the barrier.

For charges of unequal mass, Zaker observed that coalescence persists at longer delays with a weight ratio of 1:2 and vanishes at shorter delays with a weight ratio of 2:1 compared to equal charges. (Thus, the criterion given in AR 385-64 uses more conservative values of 4.0 and $5.6 \text{ ms/lb}^{1/3}$, respectively.) Zaker also performed some experiments with three equal charges. He observed that the third pulse tends to overtake the second before the second overtakes the first.

5.2 Simulation of the Experiments Using the 1-KT Standard. In order to simulate Zaker's equal-charge experiments, we made BWACO computations for two 1.31-lb "stacks" separated by 0.254 m, specifying only one possible initial donor. Specific delay times were obtained using appropriate values of the propagation velocity. The region of interest was specified as extending to 18.0 m from the center of charge in each direction, corresponding to the region of experimental observation. The delay times associated with the vanishing of coalescence at the edges of the region may be compared with experimental values.

Coalescence maps of regions of the surface for various delay times are plotted in Figure 1. The donor (solid circle) and acceptor (solid square) are plotted near the center of the map and overlap. Regions of coalescence are shaded with dots. Unshaded areas experience two independent waves, while shaded areas experience a single coalesced wave. For delays less than 3.3 ms, coalescence within the region of interest is predicted in all directions. For delays greater than 4.0 ms coalescence in the lateral (x) direction vanishes. Thus, with a delay of 4.3 ms (Zaker's limit for lateral coalescence), coalescence in the lateral direction is not predicted. Coalescence in the axial (y) direction persists until the delay exceeds 4.7 ms. This may be compared to Zaker's value of 6.1 ms. The difference between the delays for which coalescence vanishes in the axial and lateral directions respectively is only 0.7 ms, which is much less than Zaker's value of 1.8 ms.

In each direction, vanishing of coalescence is predicted at shorter delays than observed in or estimated from the Zaker experiments. The discrepancy is most significant in the axial direction. This result, which

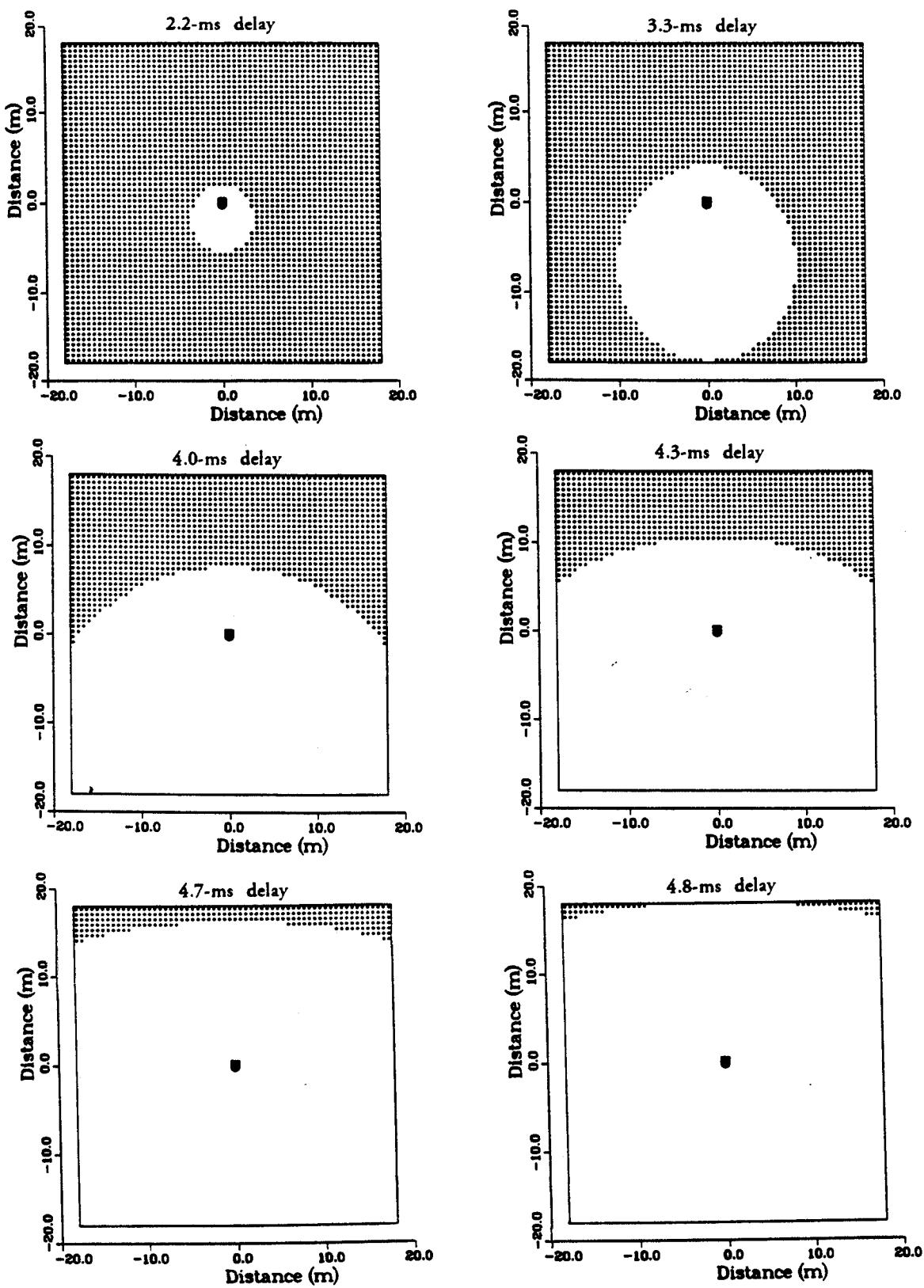


Figure 1. Coalescence maps for Zaker experiment simulations using the 1-KT standard.

is unacceptable, contradicts our expectations for the bias in BWACO's coalescence algorithm and suggested to us that the 1-KT Standard might be inaccurate.

5.3 Alternatives to the 1-KT Standard Fits. Kingery and coworkers have reported air blast parameters, including shock arrival time, positive phase duration and peak overpressure, versus distance for large-scale hemispherical TNT surface bursts (Kingery, Keefer, and Day 1962; Kingery and Pannill 1964; and Kingery 1966). This data provides a basis for evaluating the applicability of the 1-KT Standard. Comparison with the 1-KT Standard shock arrival time is shown in Figure 2. There is close agreement except in a tiny region very near the source. Comparison with the 1-KT Standard positive phase duration is shown in Figure 3. The values from the standard are generally too short at distances of interest. This would tend to suppress the prediction of coalescence, producing the results we observed. Comparison with the 1-KT Standard peak overpressure is shown in Figure 4. Zaker's small-scale results are also plotted. The 1-KT Standard pressures are generally lower than the measured values.

Alternative fits which give shock arrival time and positive phase duration as functions of distance were developed. These retain the appropriate asymptotic behavior at large distances from the source. The fit for shock arrival time is of the form

$$t_a = a_a [r - b_a \tanh(\frac{r}{r_a})],$$

where a_a , b_a and r_a are constants. The fit for positive phase duration is of the form

$$\Delta t_+ = a_+ (1 - \frac{b_+}{r^{c_+}}) [\ln(r^2 + r_+^2)]^{1/2},$$

where a_+ , b_+ , c_+ and r_+ are constants. This fit is not used for very small values of r , for which the value of the positive phase duration is held constant. Kingery and Pannill (1964) provided a fit for peak overpressure. These fits are also plotted in Figures 2 through 4 and provide improved agreement with measurements in each case.[†]

[†]After implementation of this standard, we learned of the existence of a TNT standard.

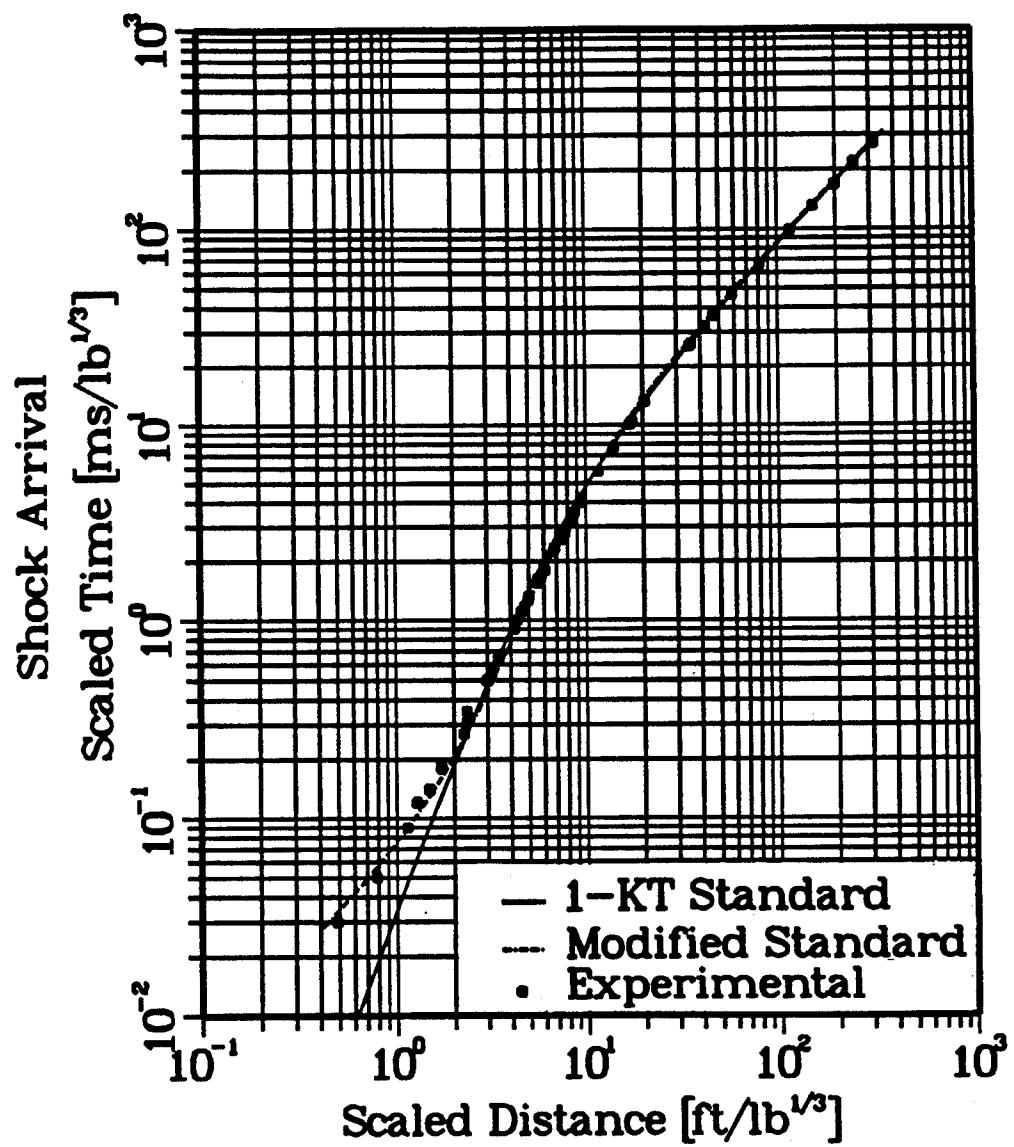


Figure 2. Comparison of the 1-KT standard with the alternative standard for shock arrival time.

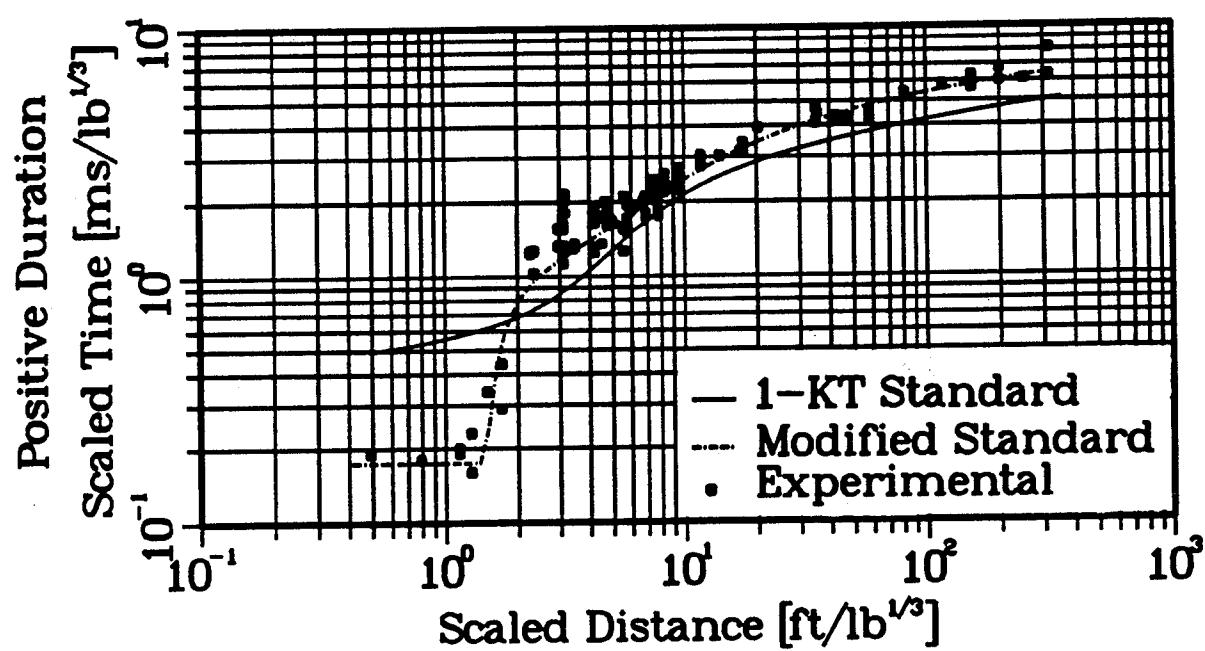


Figure 3. Comparison of the 1-KT standard with the alternative standard for positive phase duration.

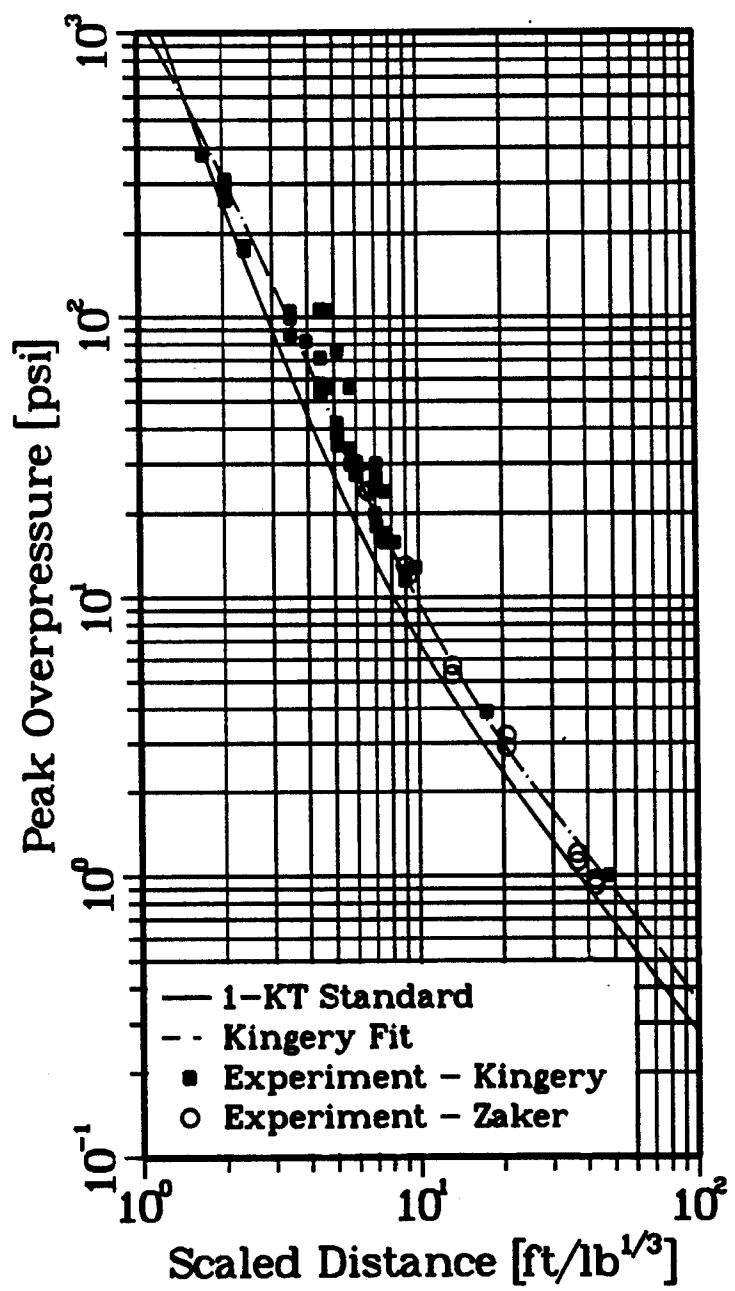


Figure 4. Comparison of the 1-KT standard and the Kingery fit with the Kingery and Zaker data for peak overpressure.

5.4 Simulation of the Experiments Using the Alternative Standard. Coalescence maps obtained using the alternative standard are shown in Figure 5. Lateral coalescence vanishes for delays exceeding about 5.1 ms, which is significantly greater than Zaker's observation of 4.3 ms. In the axial direction, coalescence vanishes for delays of 5.9 ms or greater. This is only slightly less than Zaker's estimate of 6.1 ms. There is no significant change in the difference between the axial and radial values. It is still less than half that determined by Zaker. The predictions obtained with the alternative standard do not miss significant regions of coalescence and are acceptable.

As we have noted, at sufficiently large distances from the stacks, coalescence will be predicted. For the present conditions, this becomes apparent upon expanding the regions of interest while maintaining the other parameters as shown in Figure 6. Although coalescence is identified at large distances, the pressures produced are too low to be of concern. BWACO was configured to suppress identification of coalescence where the combined peak overpressure is below 0.9 psi (the maximum permissible exposure level for inhabited buildings subjected to blast waves from large charges). The effect on the coalescence maps is to eliminate shading where distance from the charges is sufficiently large as illustrated in Figure 7. It appears that the pressures combined using simple superposition are somewhat high, as the extent of the region of significant pressure is greater than that associated with a single charge of the same total weight. As a result of the expanded extent, coalescence is observed at somewhat longer delays than those associated with the experimental region of significant pressure specified in the earlier computations. (See Figure 5.)

5.5 Effect of Pressure Combination Algorithms. To assess the effect of the pressure combination algorithms on the extent of the region of significant pressure, we made computations with from 2 to 10 stacks, located at the same point, having the same total equivalent explosive weight (2.62 lb). The ratio of the radius of the region of significant pressure produced by the detonation of n stacks to that produced by the detonation of one stack is plotted as a function of n in Figure 8. Contrary to Zaker's observation, using simple superposition, we obtain substantial increase in the extent of the region of significant pressure with increasing number of stacks even though the total explosive weight remains constant.

The results obtained with the LAMB algorithm (also shown in Figure 8) are virtually identical to those obtained using simple superposition. The LAMB algorithm has been shown to produce correct results for the head-on collision of two equal blast waves and overestimates the pressure in other cases (Hikida and

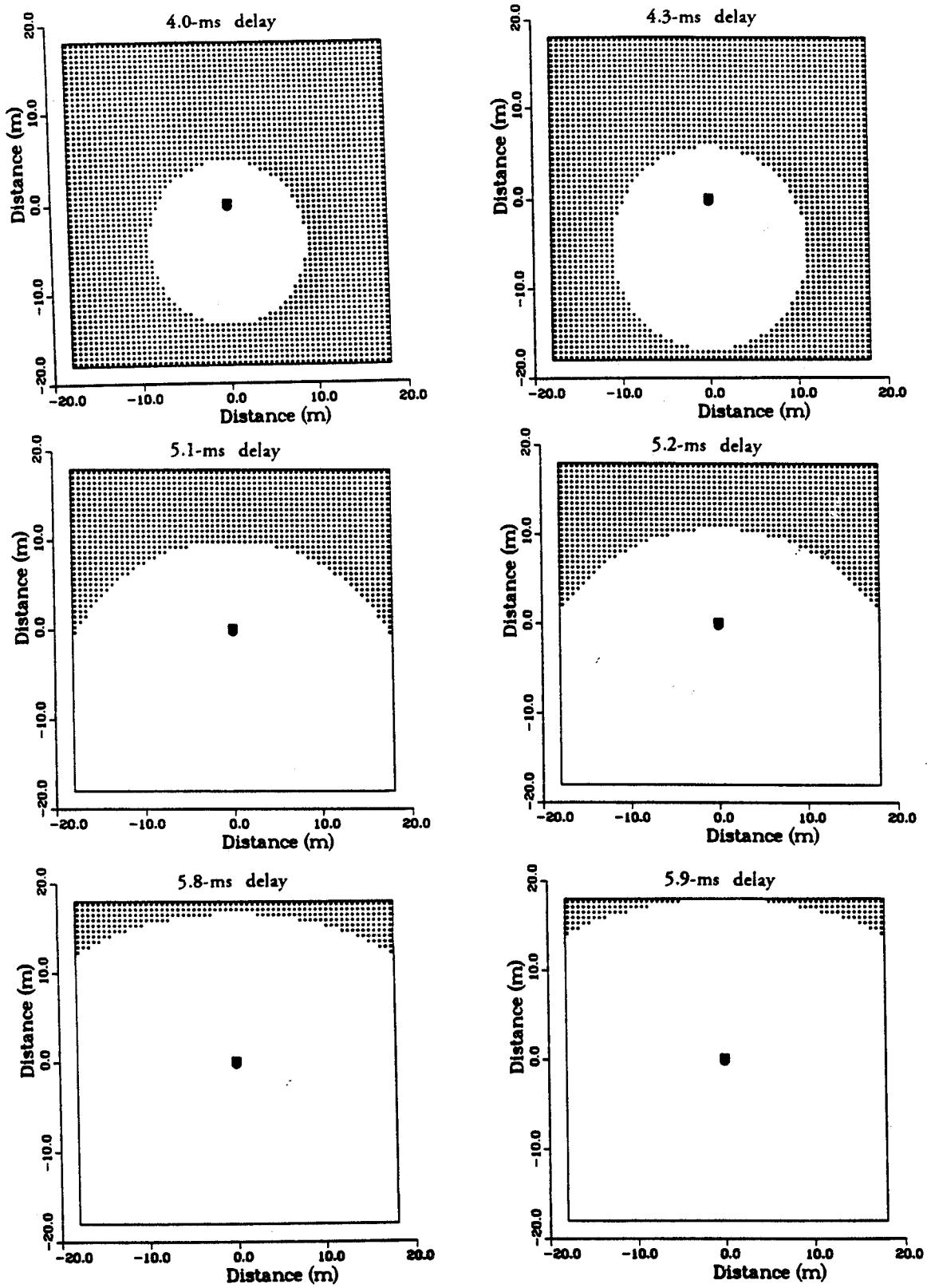


Figure 5. Coalescence maps for Zaker experiment simulations using the alternative standard.

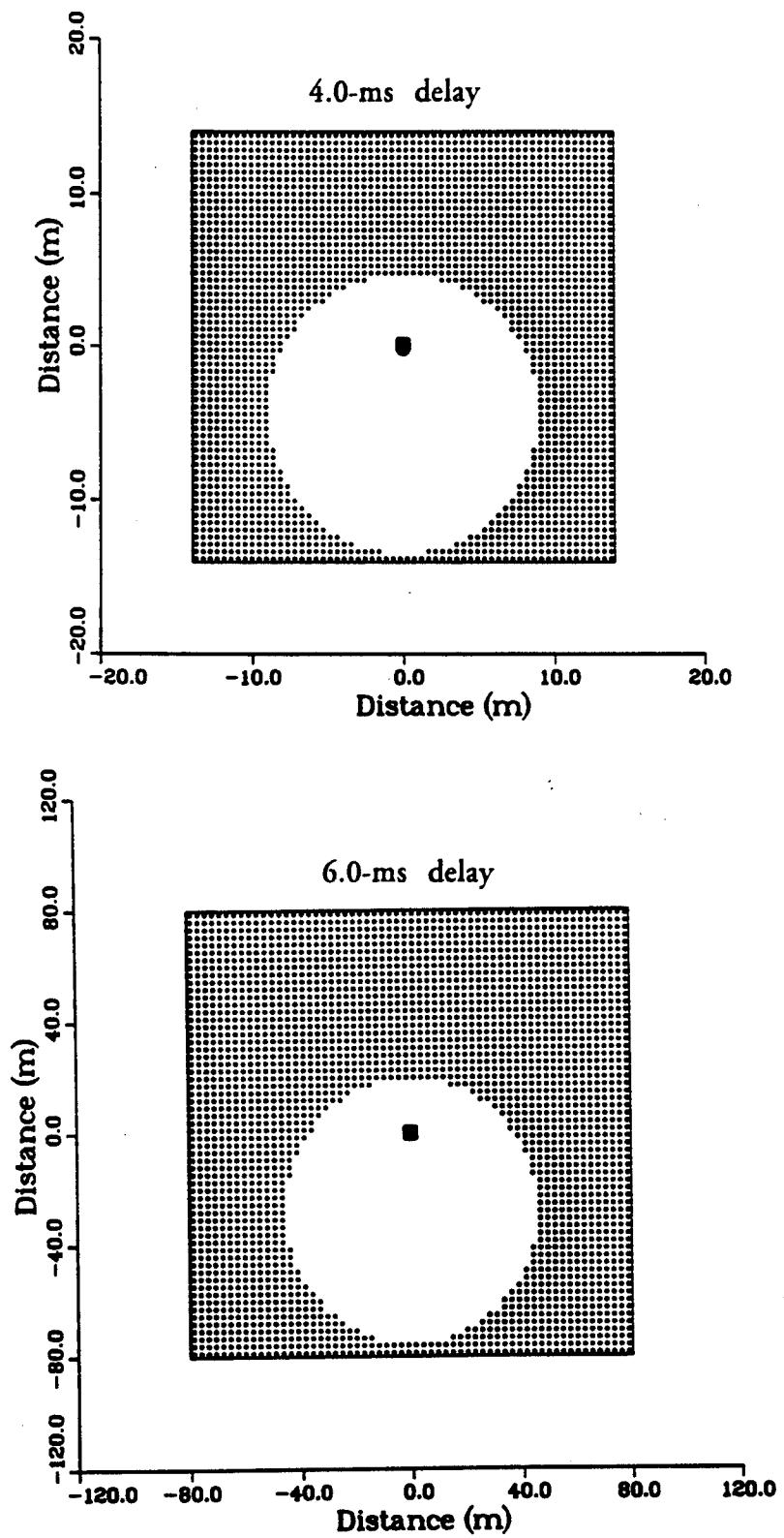


Figure 6. Coalescence maps showing the effect of scale on coalescence at two different delay times.

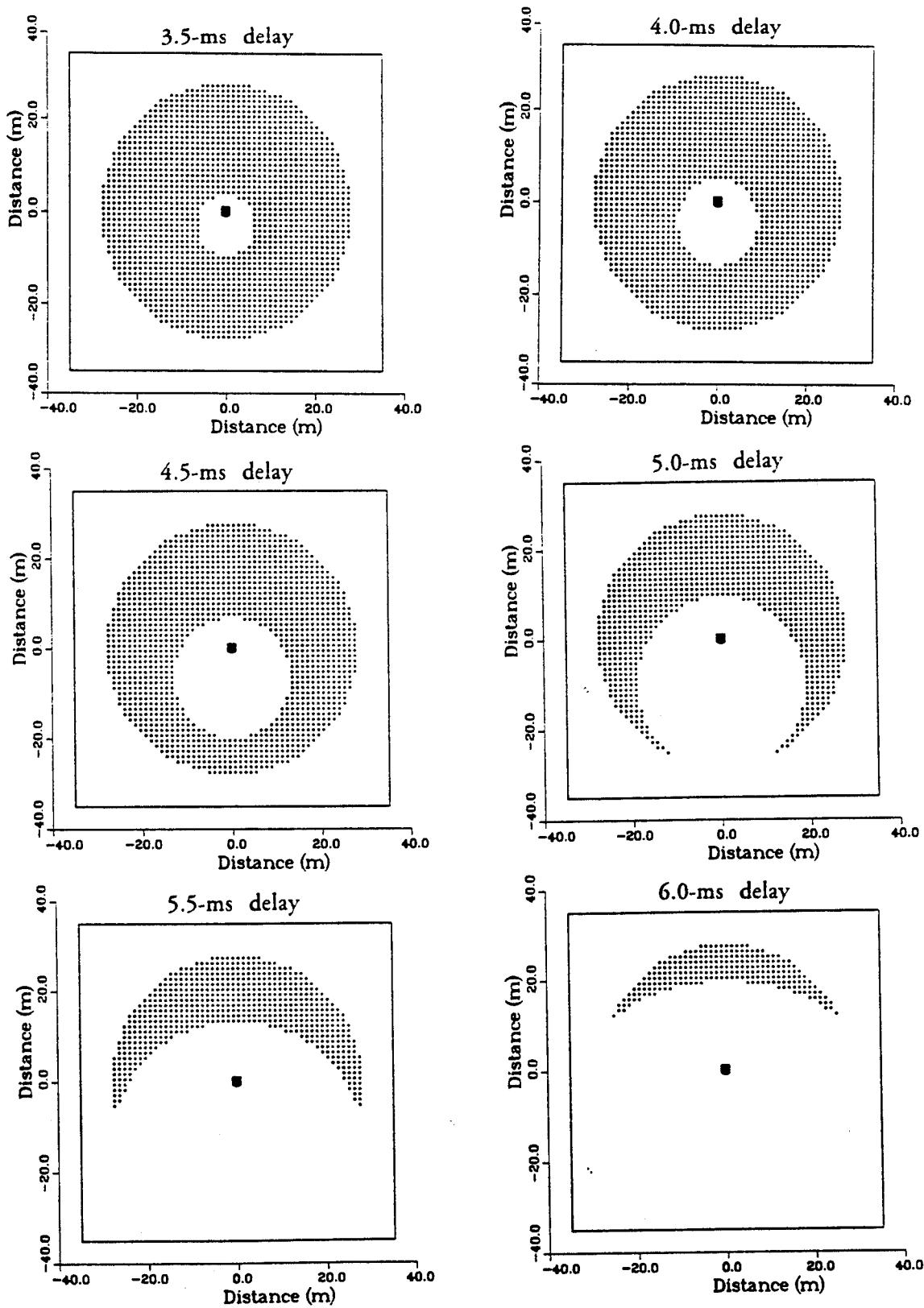


Figure 7. Coalescence maps for Zaker experiment simulations with coalescence indications suppressed at low peak overpressure.

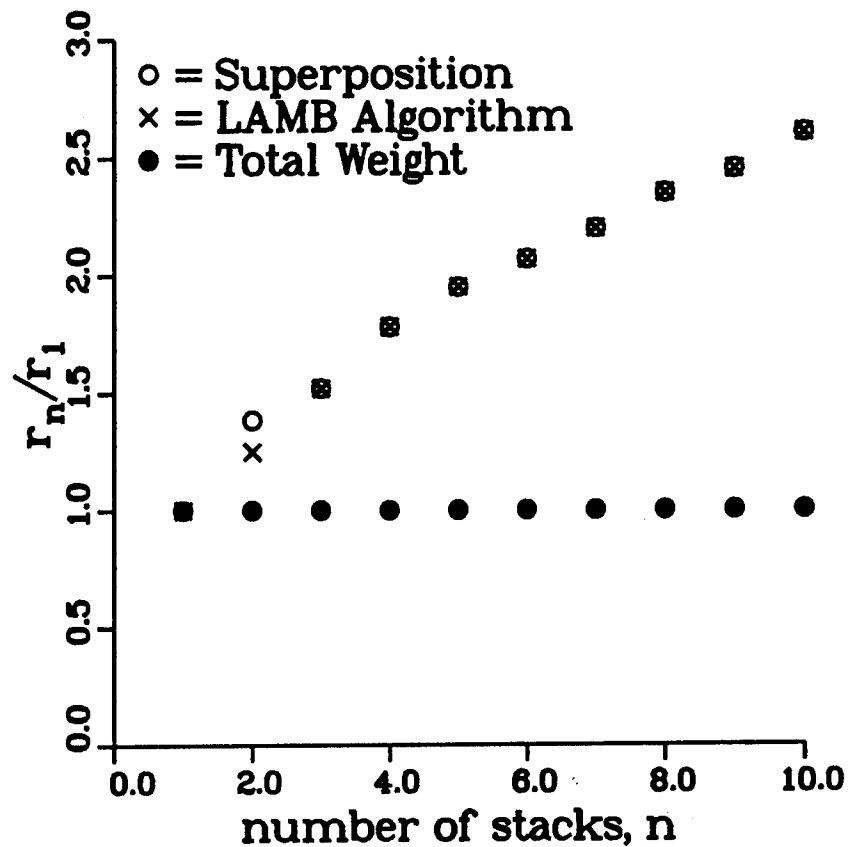


Figure 8. Comparison of the effects of the pressure combination algorithm on the region of significant pressure as a function of the number of charges.

Needham 1981; Brode 1977). In the case of overtaking waves, the magnitude of the overestimate appears significant.

As an alternative to the simple superposition and LAMB algorithms, we computed the combined pressure by first combining the stacks which produce the coalesced wave into a single stack positioned at the center of charge. This approach, suggested by Zaker's observation, is applicable where the angles between coalescing waves are small, as occurs at any point whose distance from the stacks significantly exceeds the distance between stacks. It produces regions of significant overpressure which do not increase with the number of stacks. This result is also shown in Figure 8.

5.6 Simulation of the Experiments Using the Alternative Standard and Pressure Combination Algorithm

The way in which the alternative pressure combination algorithm modifies the results obtained with two equal charges is shown in Figure 9. As expected, the region of significant pressure is reduced in comparison to that shown in Figure 7. The delays at which coalescence vanishes are slightly increased compared with those associated with the analysis shown in Figure 5 because the region of significant pressure is larger than the region of experimental observation.

We also made computations with unequal charges in ratios of 1:2 and 2:1 with a constant total equivalent weight of 2.62 lb. Figure 10 shows coalescence maps for the 1:2 case at delays ranging from 4.0 to 6.5 ms. There is still a small region of coalescence at the longest delay. Figure 11 shows coalescence maps for the 2:1 case at delays ranging from 3.0 to 5.5 ms. In this case, coalescence has completely vanished at the longest delay. Figure 12 compares results from each of these cases with the equal-charge case at two different delays. This shows that the persistence of coalescence decreases as the ratio of the donor weight to the acceptor weight increases. The result is consistent with Zaker's observations.

Figure 13 shows coalescence maps obtained in a simulation of Zaker's three-charge experiment. The size of each shading dot is proportional to the number of waves coalesced at the corresponding station. However, it cannot be ascertained which waves have coalesced. For short delays, coalescence of all three waves is predicted in all directions, although two-wave coalescence occurs first (closer to the center of charge). As the delay is increased, three-wave coalescence begins to vanish, starting in the negative axial direction. Two-wave coalescence persists for longer delays. The results are consistent with Zaker's observation that the third pulse tends to overtake the second before the second overtakes the first.

5.7 Results With Multiple Initial Donors

To this point, all the computations have specified a single donor. Each resultant coalescence map shows the coalescence pattern associated with a single chain of events. However, BWACO is generally intended to be used with all stacks specified as possible initial donors. The coalescence map produced in this case is a composite which reflects the worst case loading at each point in the region of interest. The effect of allowing each stack to act as an initial donor in the Zaker simulations was examined in order to check the code's multiple donor algorithm.

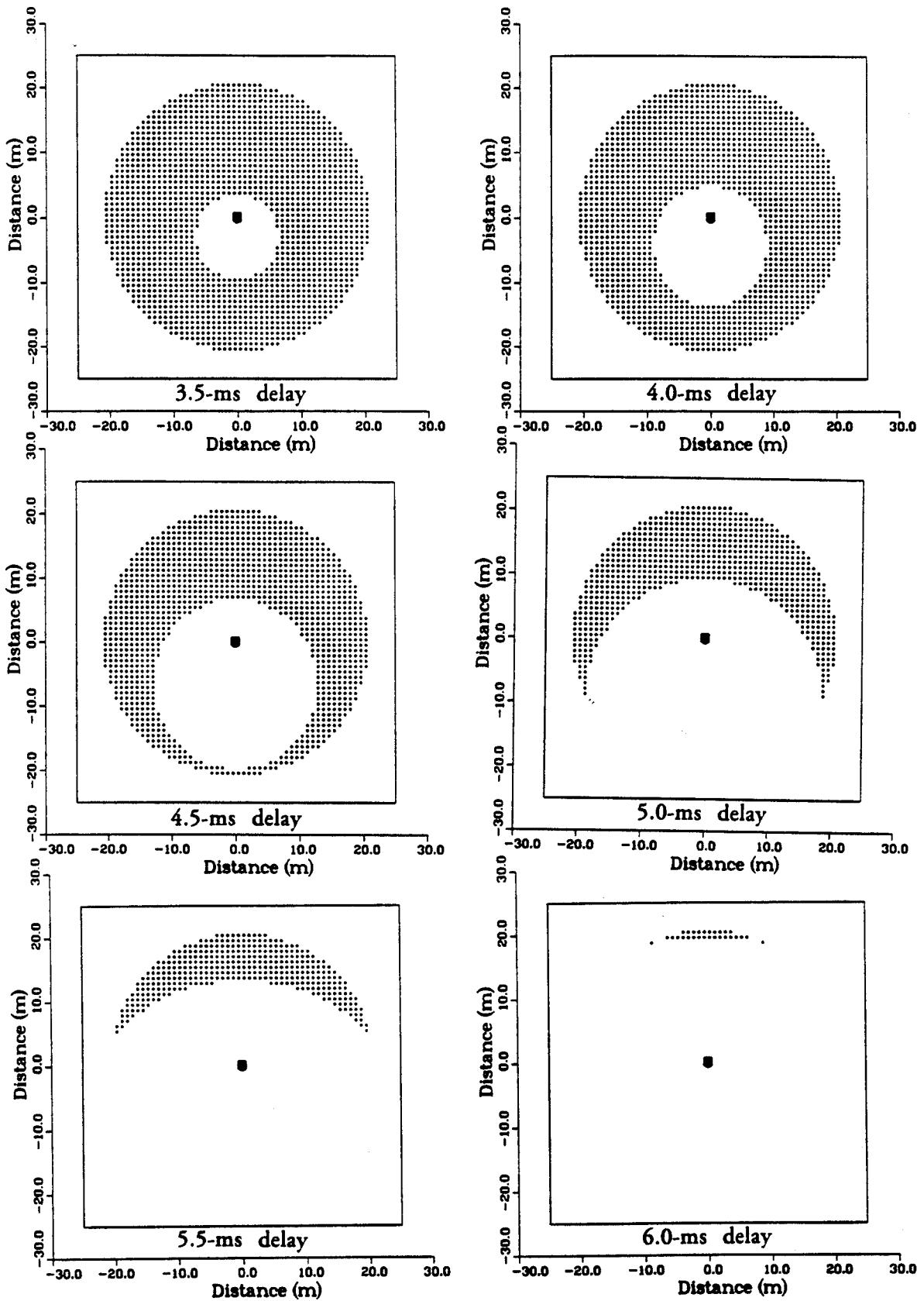


Figure 9. Coalescence maps for Zaker experiment simulations using the alternative standard and pressure combination algorithm.

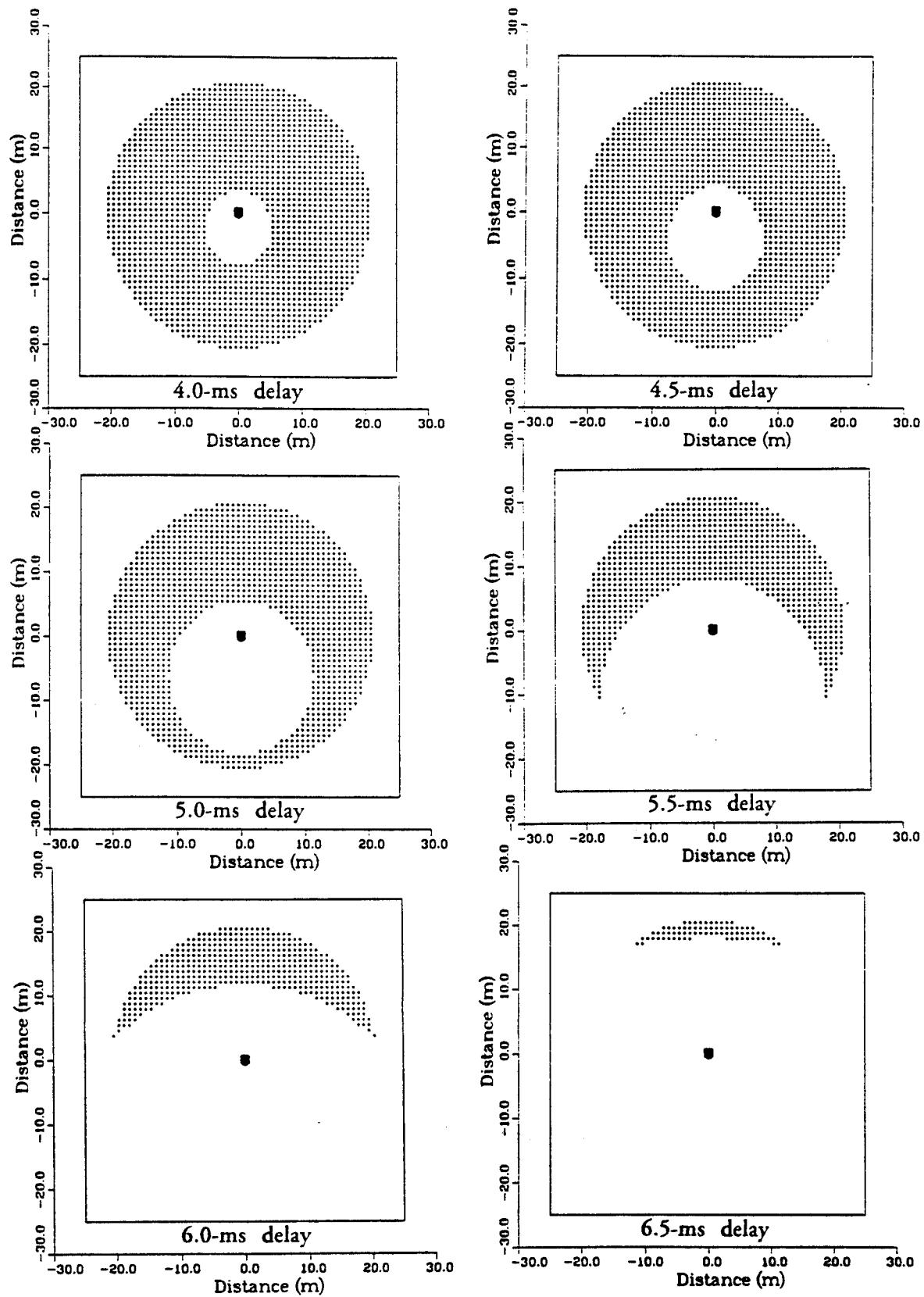


Figure 10. Coalescence maps for Zaker experiment simulations with unequal charges having weight ratios of 1:2.

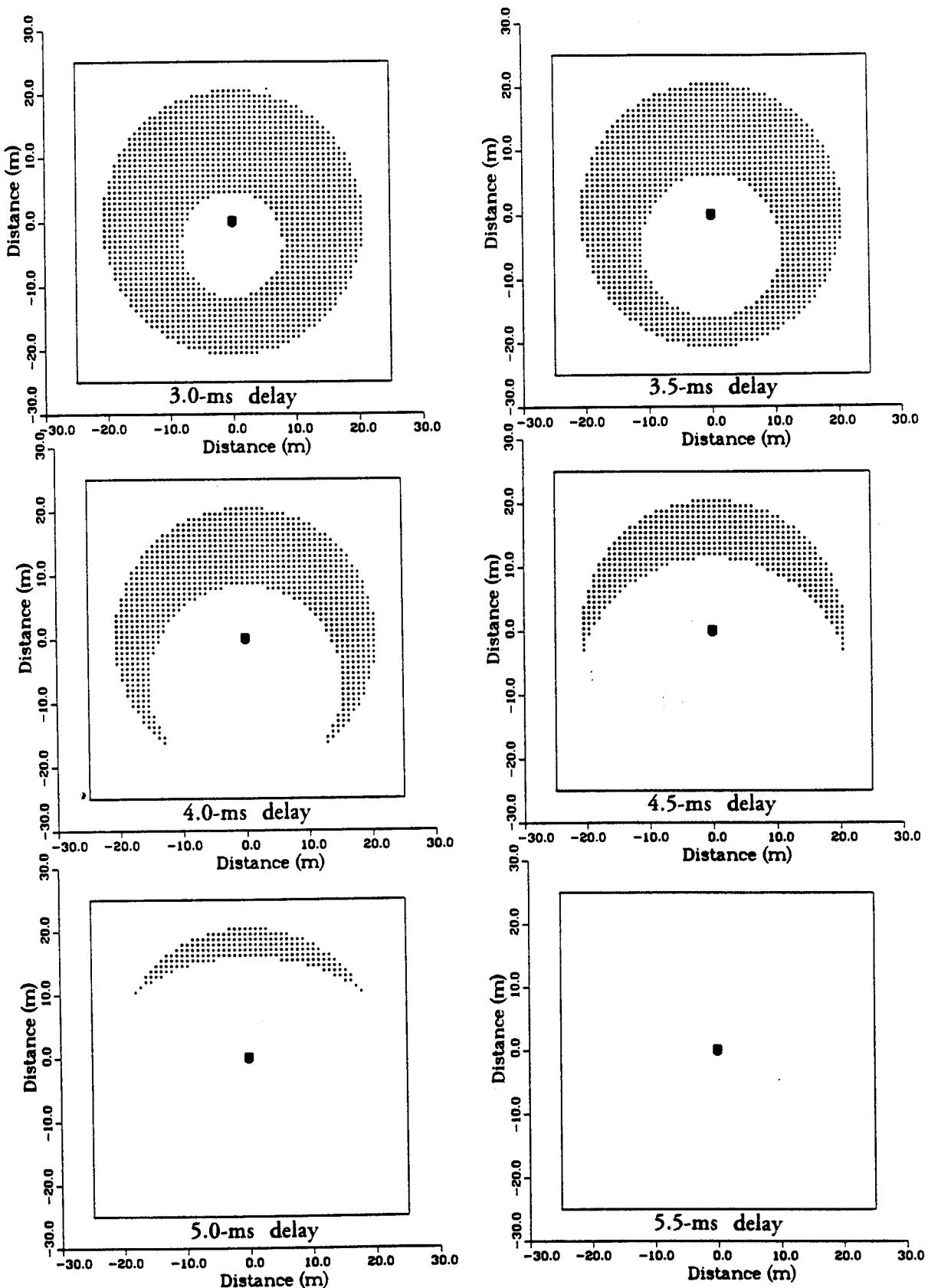


Figure 11. Coalescence maps for Zaker experiment simulations with unequal charges having weight ratios of 2:1.

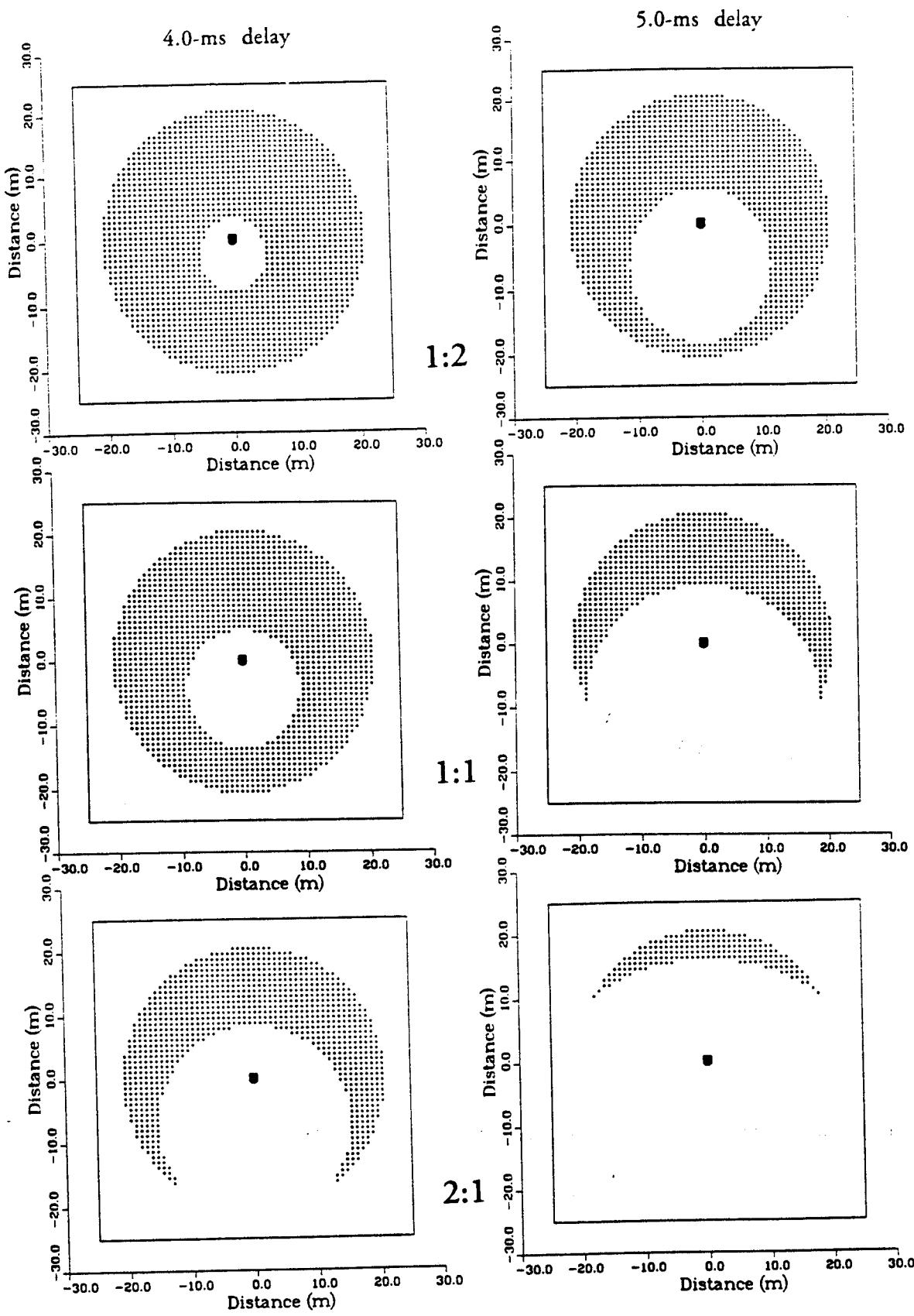


Figure 12. Coalescence maps for Zaker experiment simulations with unequal charges having weight ratios of 1:2 and 2:1.

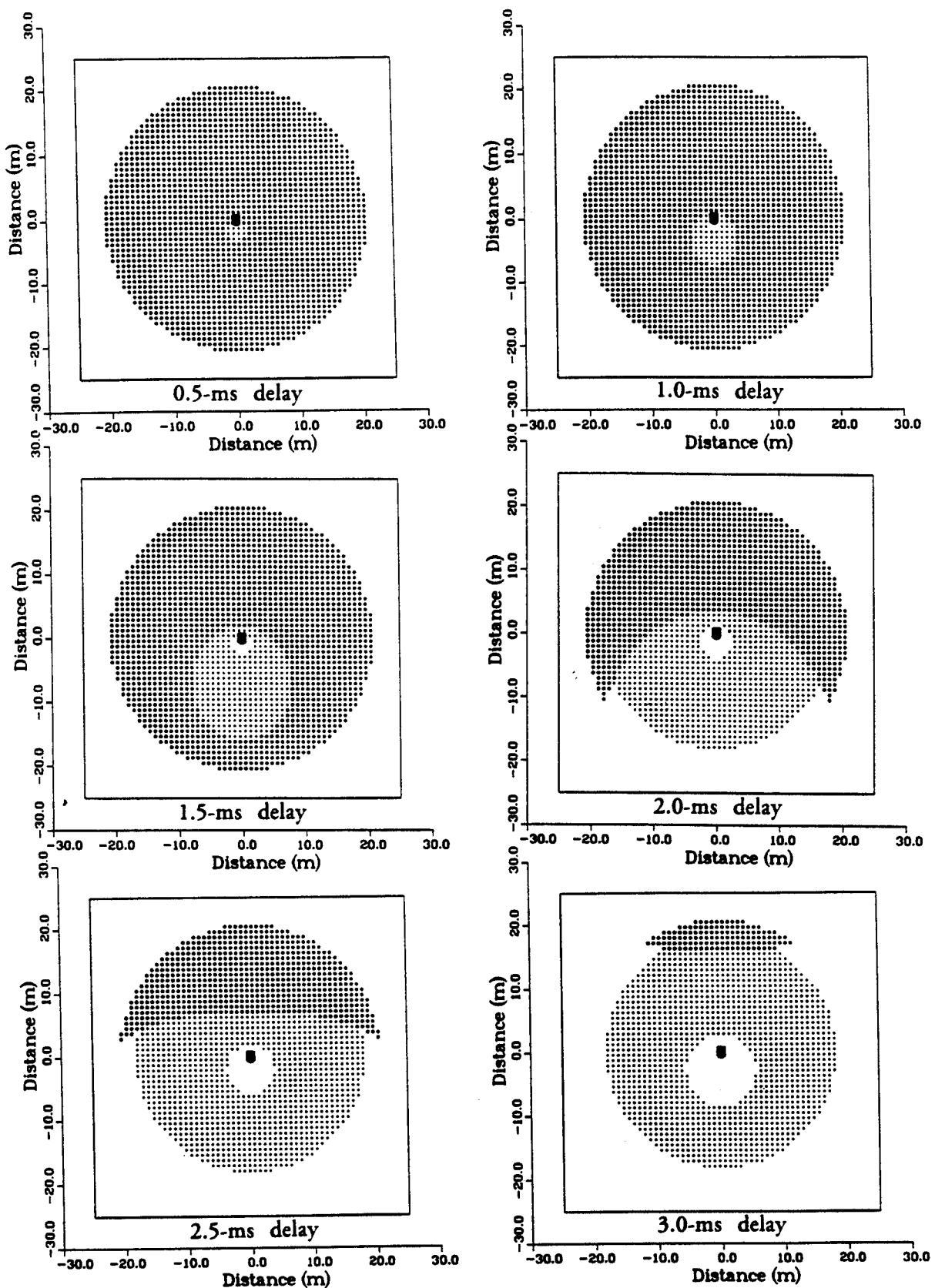


Figure 13. Coalescence maps for Zaker experiment simulations with three equal charges.

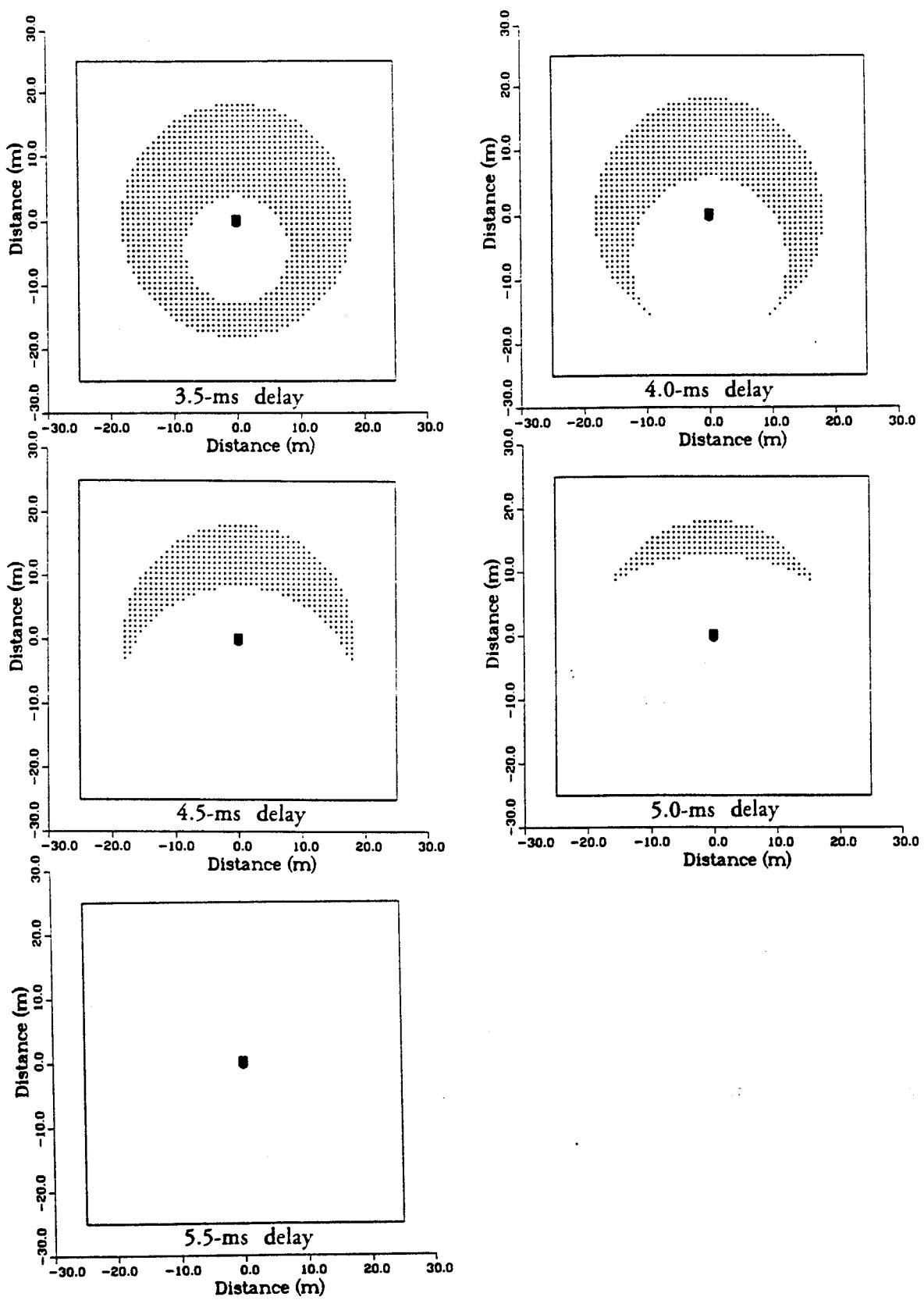


Figure 13. Coalescence maps for Zaker experiment simulations with three equal charges (continued).

The coalescence patterns produced in simulations of two equal charges are shown in Figure 14. As expected, symmetry is produced, with the single-donor coalescence regions (shown in Figure 9) reflected across the horizontal axis.

The coalescence regions produced with two unequal charges are not symmetrical about the horizontal axis as shown in Figure 15. These are combinations of the single-donor regions (shown in Figure 10) with reflections of the single-donor regions (shown in Figure 11) about the horizontal axis.

Symmetry about the horizontal axis also results with three equal charges, as shown in Figure 16. In this case, however, the region of coalescence is greater than that associated with reflection of the single donor regions (shown in Figure 13) because coalescence produced by the center-charge donor dominates. In fact, coalescence of the two waves from the outer charges persists at all delays since they detonate simultaneously when the central charge is the donor.

6. PERSONAL COMPUTER VERSION GRAPHICS

When BWACO was adapted for use on the personal computer, the coalescence map was enhanced with the addition of contours of the total explosive weight associated with the coalesced waves and a capability for producing peak overpressure contour plots was added. Examples of these maps are shown in Figure 17. The maps may be obtained either on the PC display or in hard-copy form.

Regions in which coalescence is detected are hatched in the coalescence map. Each contour is labeled to indicate the total explosive weight contributing to the worst-case blast environment within that contour. However, the outermost contour line is further limited by the overpressure limit for inhabited buildings. An inhabited building distance circle centered at the center of the charge is also shown. The legend lists all the contour levels present in the plot. The highest contour level represents the maximum effective explosive weight for the configuration.

In the peak overpressure map, contours are labeled to indicate the worst-case peak overpressure at each point within the region of interest. The contour levels are those specified in AR 385-64. The lowest level varies between 0.9 and 1.2 psi, depending on the total explosive weight. The limits of the blast hazard area determined in accordance with the regulations are also shown. The legend lists the stack positions

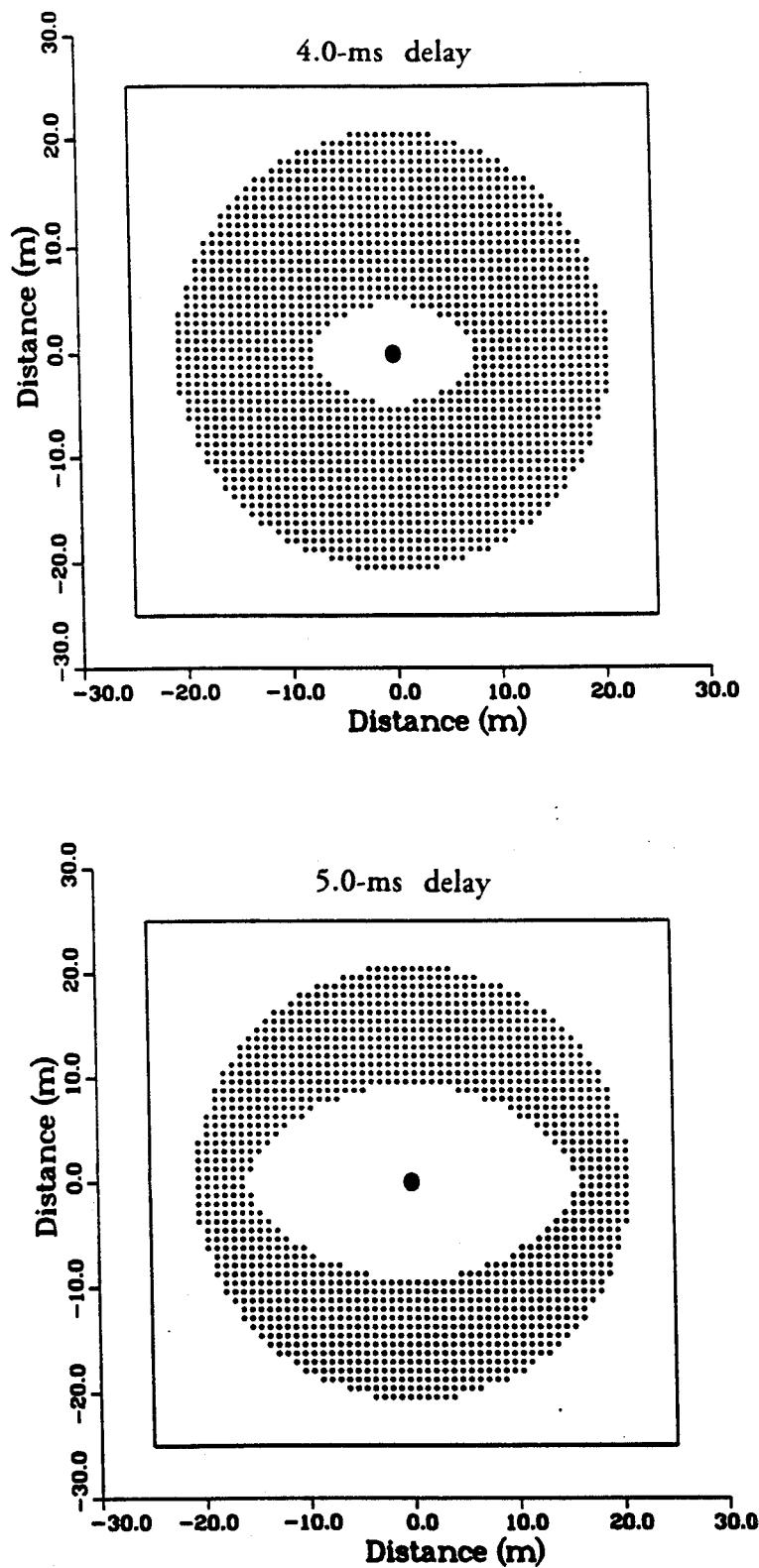


Figure 14. Coalescence maps for Zaker configuration simulations with two equal charges and multiple donors.

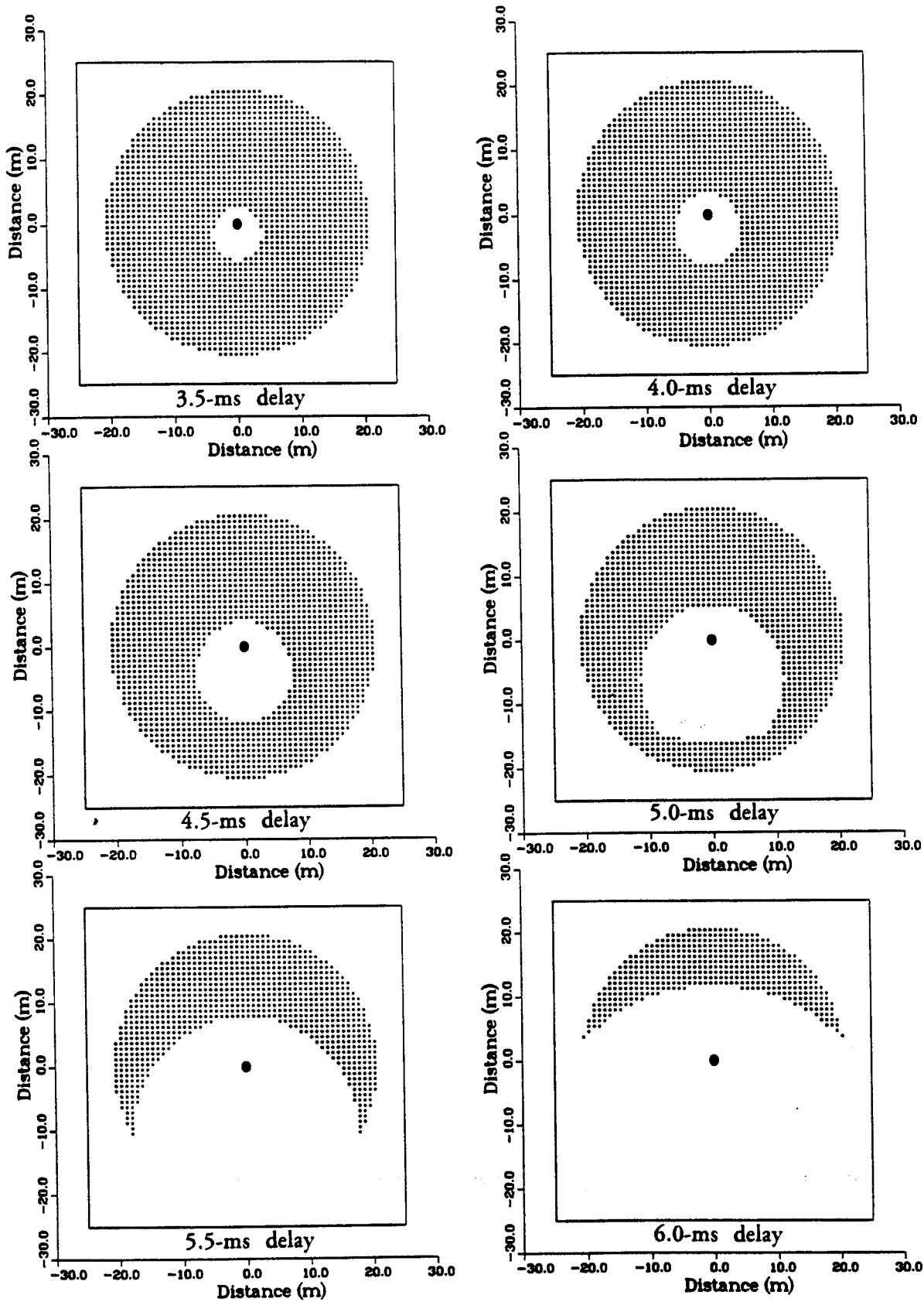


Figure 15. Coalescence maps for Zaker configuration simulations with two unequal charges and multiple donors.

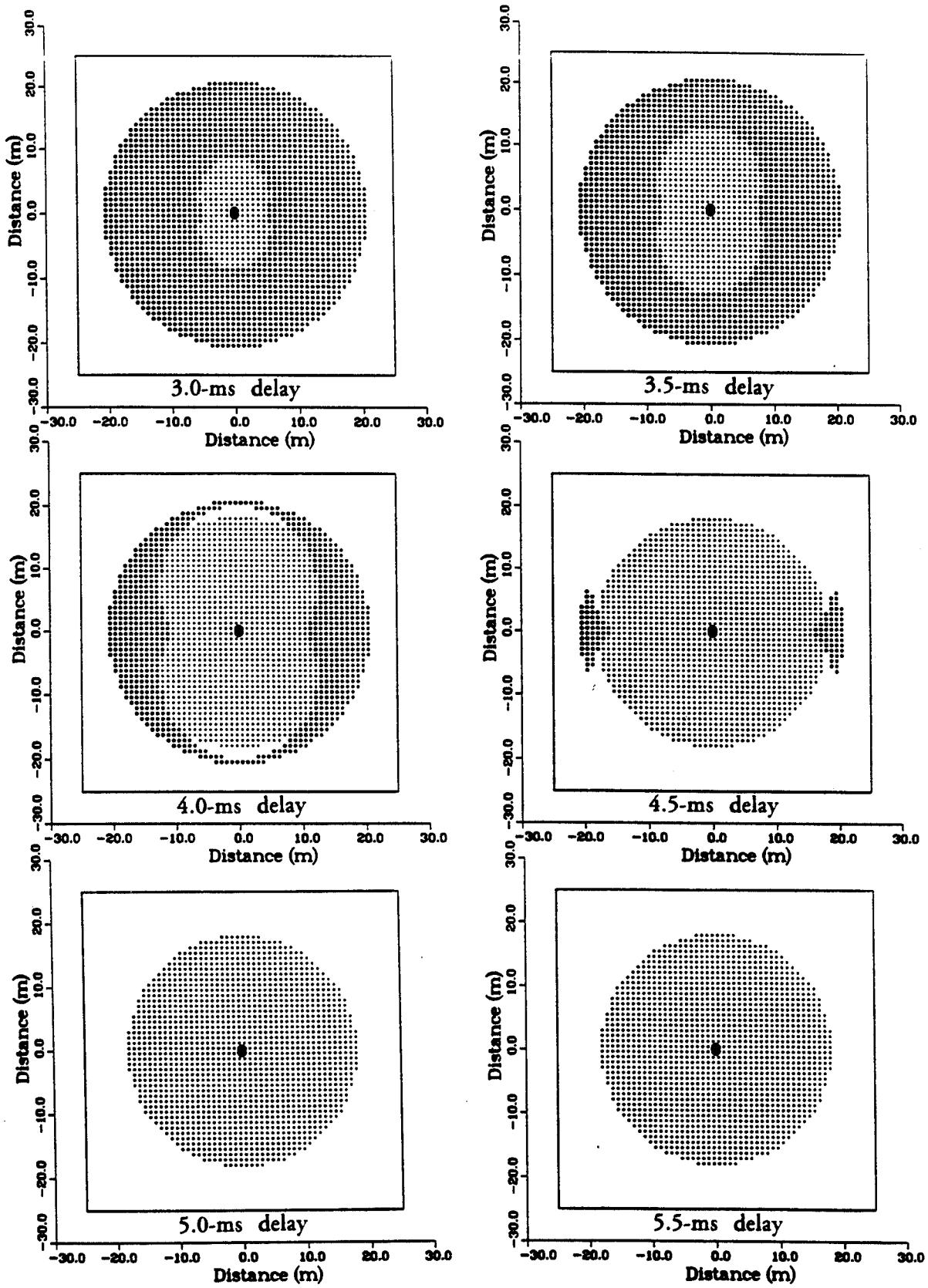


Figure 16. Coalescence maps for Zaker configuration simulations with three equal charges and multiple donors.

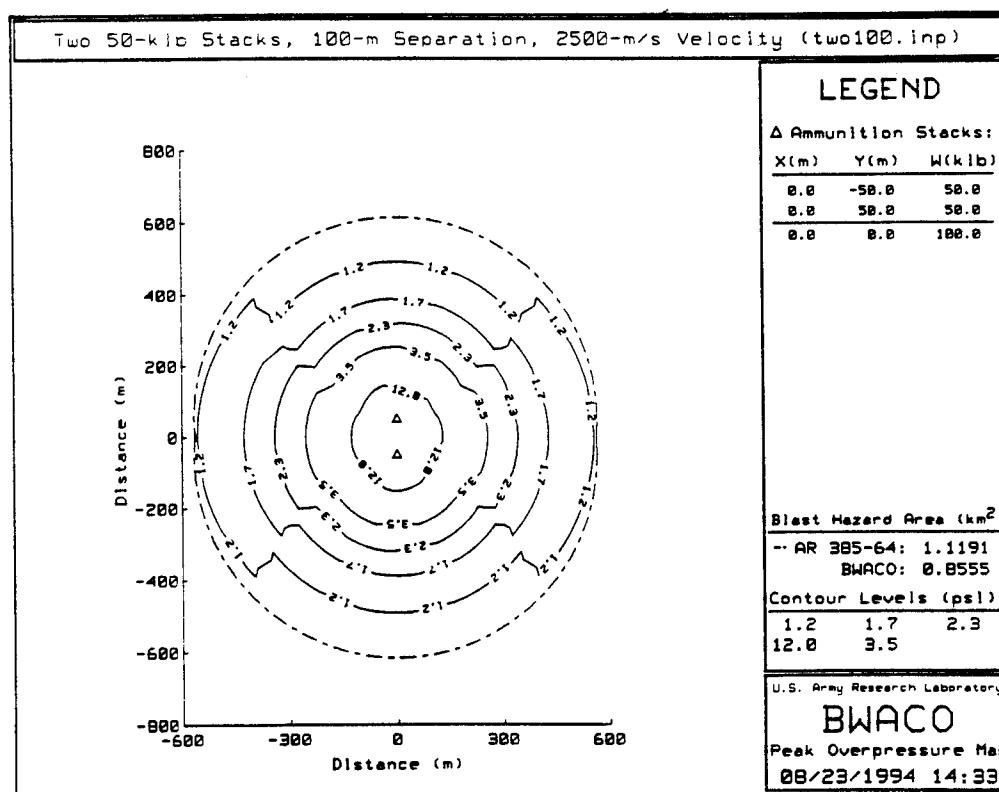
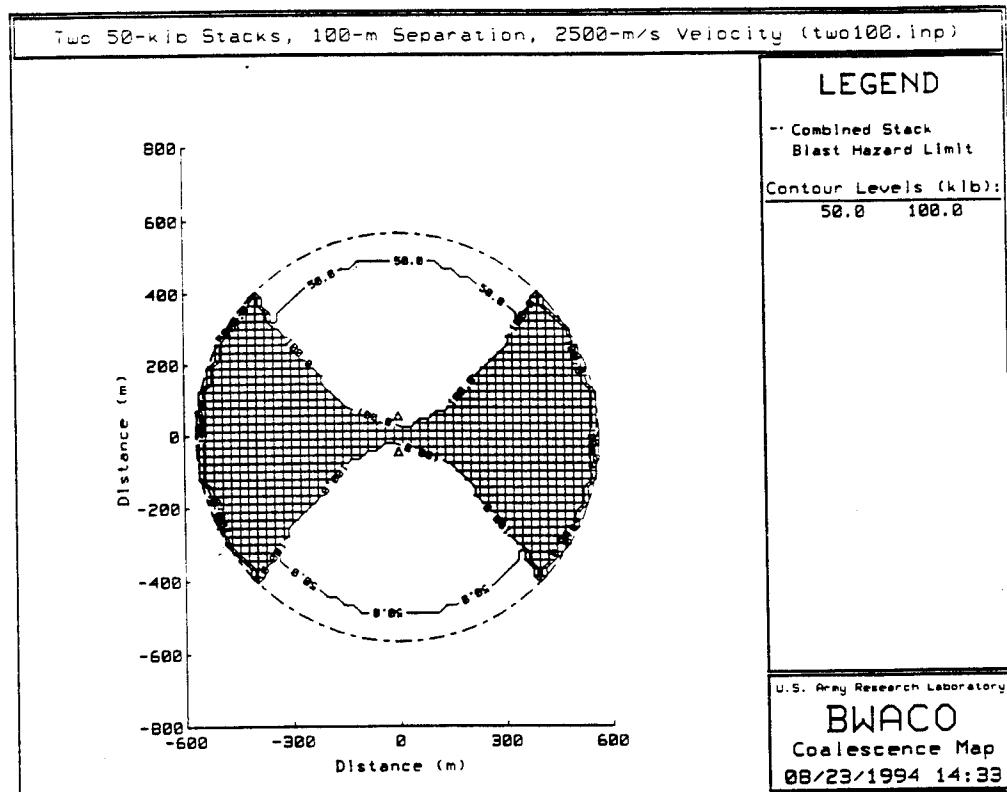


Figure 17. Coalescence and peak overpressure maps from the PC version of BWACO.

and weights, along with center of charge and total weight, compares the regulatory blast hazard with that determined by BWACO, and lists the contour levels.

7. SIMULATIONS OF REPRESENTATIVE LARGE-SCALE CONFIGURATIONS

In the large-scale scenarios that BWACO is intended to treat, initiation delay depends on the separation between stacks. The simulations described in the following paragraphs were intended to illustrate the effects of the separation between stacks and the distribution of explosive weight among stacks when all stacks act as initial donors. Most of them were made using a total equivalent weight of 100,000 lb distributed over two or more stacks. For this total weight, regulations permit an inhabited building exposure of 1.2 psi or less. The velocity of propagation between stacks was fixed at 2,500 m/s.

Results obtained with two stacks of 50,000 lb each are shown in Figures 18 and 19. At the smallest separation distance (50 m), the regulatory blast hazard region nearly coincides with the BWACO prediction. As the separation is increased, significant regions in which the pressure is below 1.2 psi appear. In these cases, requiring consideration of the blast produced by the total explosive weight of both stacks is clearly too restrictive. At the larger separation distances, regions of coalescence persist only near the horizontal symmetry axis. Here, consideration of the total weight is necessary. The variation of the blast hazard areas with separation is summarized in Table 1. The regulatory hazard area increases with separation while that predicted by BWACO is minimized somewhere between 100 and 150 m.

Table 1. Variation of Blast Hazard Areas with Separation for Two 50,000-lb Stacks

Separation (m)	Blast Hazard Area	
	AR 385-64 (km ²)	BWACO (km ²)
50	1.06	0.97
100	1.12	0.86
150	1.18	0.86
200	1.23	0.88

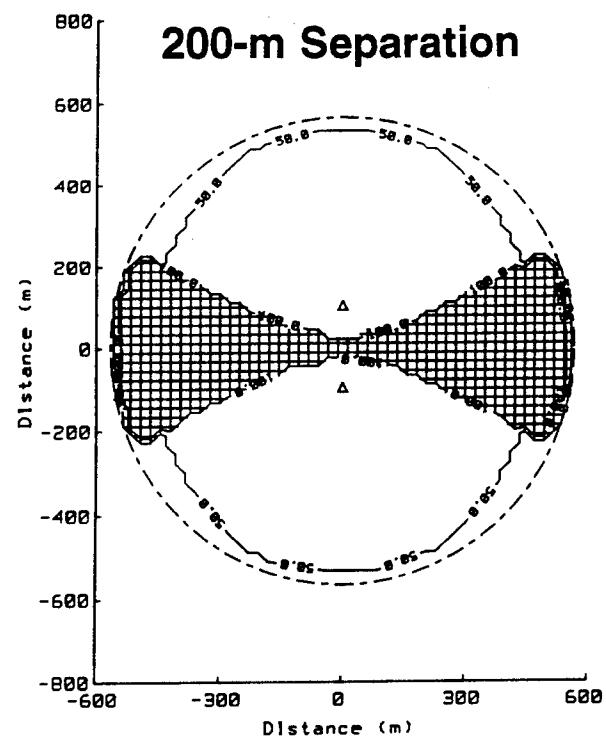
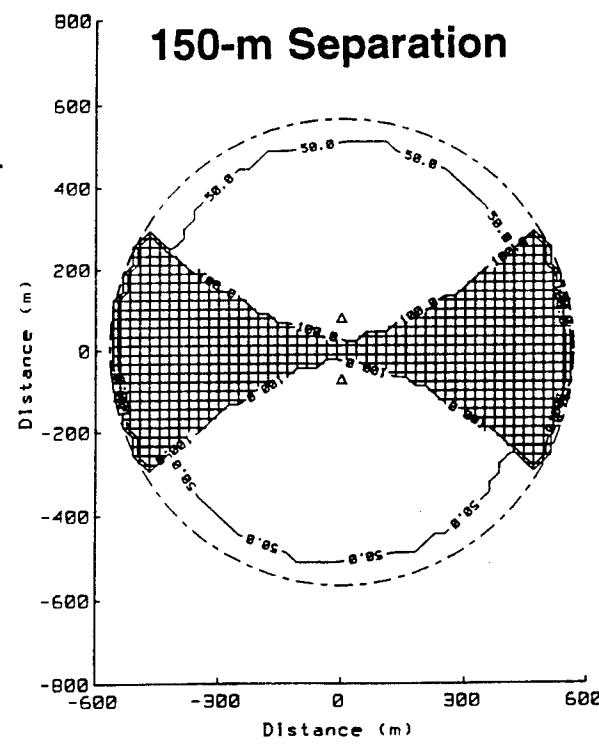
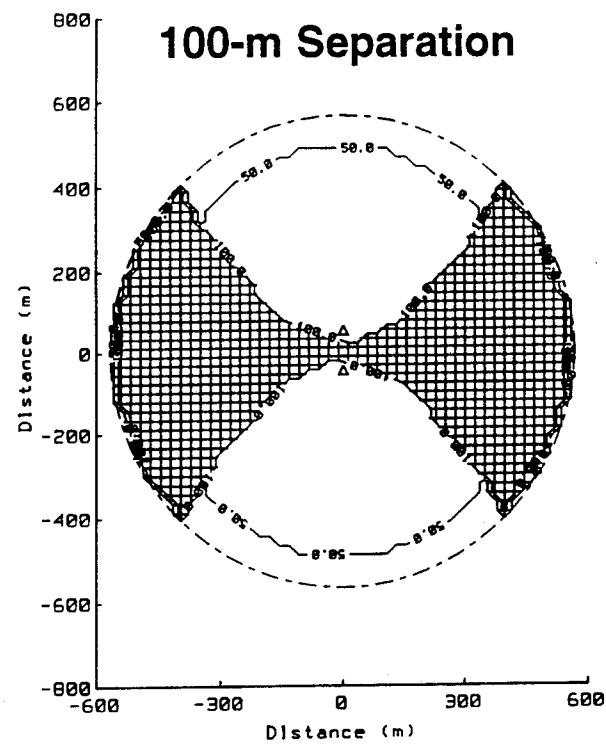
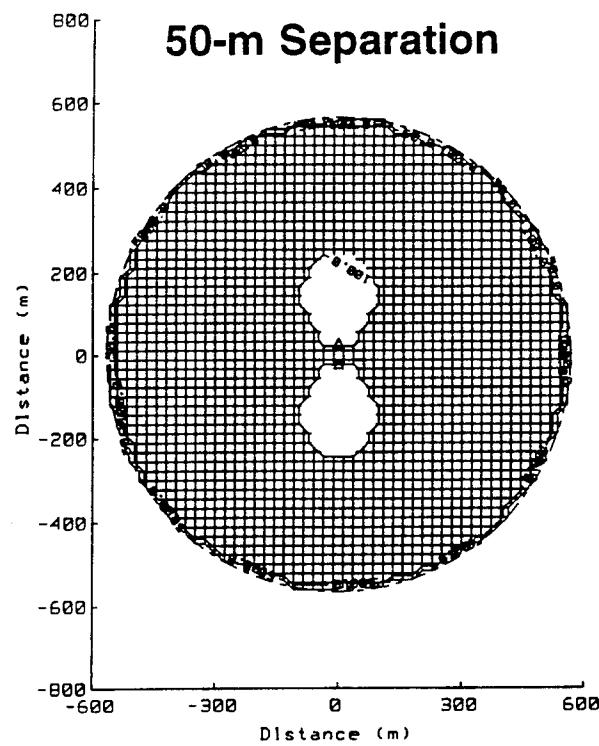


Figure 18. Coalescence maps for large-scale simulations with two 50,000-lb stacks.

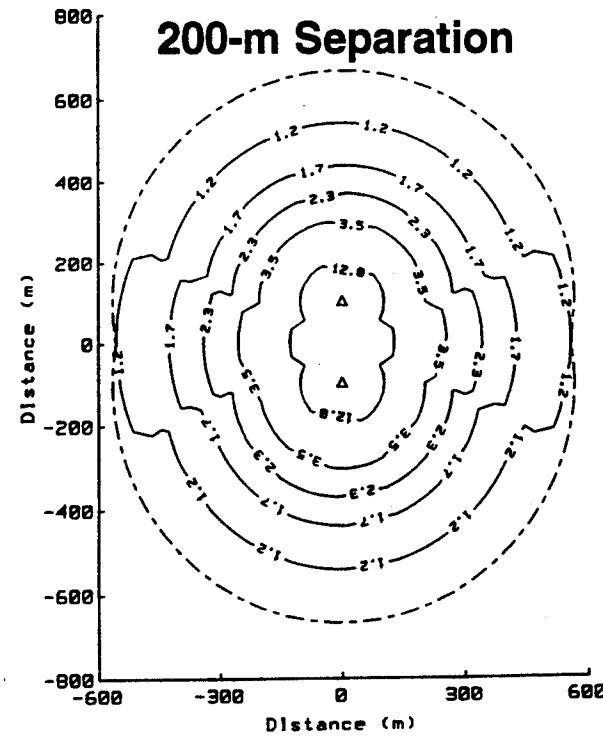
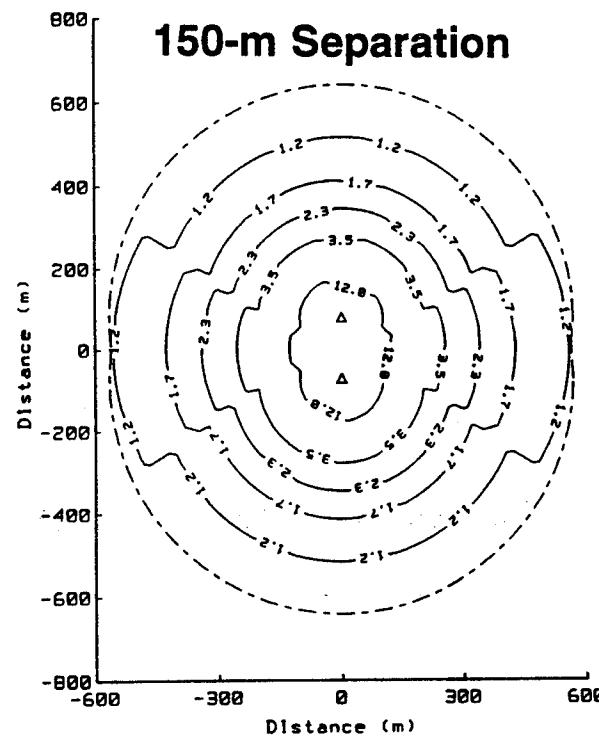
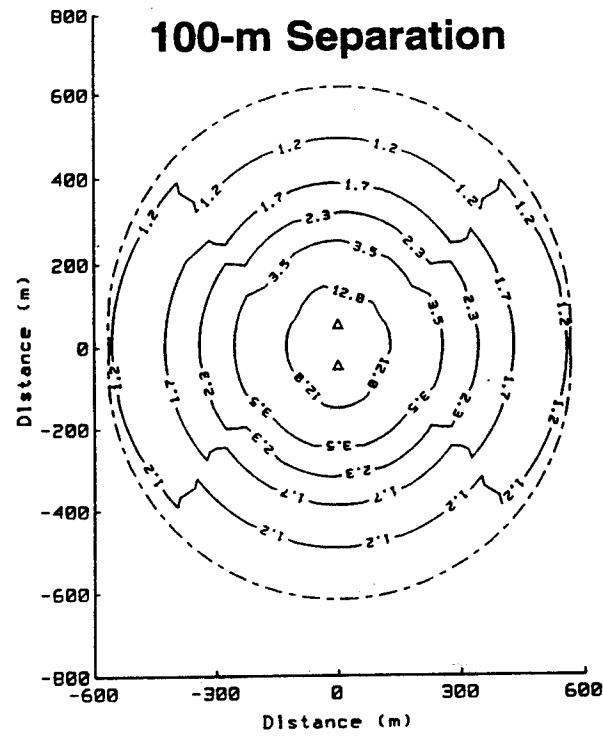
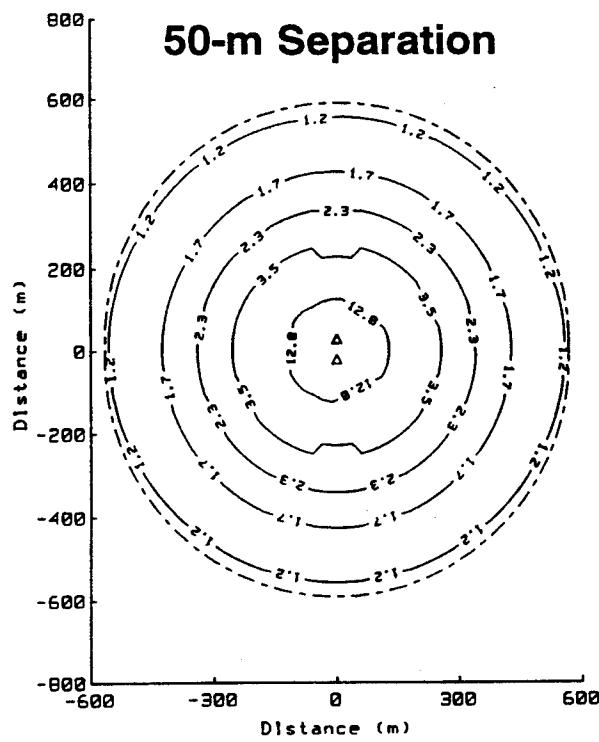


Figure 19. Peak overpressure maps for large-scale simulations with two 50,000-lb stacks.

We also simulated the effects of varying the distribution of explosive weight between two stacks separated by 150 m. The results, shown in Figures 20 and 21, indicate very little change in the coalescence region along the horizontal axis as the weight is redistributed. The hazard area increases slightly for increasingly unequal weight distributions as shown in Table 2.

Table 2. Variation of Blast Hazard Areas with Explosive Weight Distribution for Two Stacks Having a Total Weight of 100,000 lb Separated by 150 m

Weight Ratio (lb:lb)	Blast Hazard Area	
	AR 385-64 (km ²)	BWACO (km ²)
50,000 : 50,000	1.18	0.86
40,000 : 60,000	1.18	0.85
30,000 : 70,000	1.18	0.85
20,000 : 80,000	1.18	0.89
10,000 : 90,000	1.18	0.93

The results observed with four 25,000-lb stacks are similar to those seen with two 50,000-lb stacks. They are shown in Figures 22 and 23. Coalescence of waves from all four stacks and from three of the four stacks is only exhibited at the 50-m separation distance. These regions appear as small islands lying beyond the general regions of coalescence. At the 100-m separation, only two-stack coalescence results. As the separation is further increased, this coalescence becomes limited to the vicinity of the vertical and horizontal symmetry axes. The effect of separation on the blast hazard areas is given in Table 3. Again, BWACO predicts a minimum hazard near the 100-m separation.

Table 3. Variation of Blast Hazard Areas with Separation for Four 25,000-lb Stacks

Separation (m)	Blast Hazard Area	
	AR 385-64 (km ²)	BWACO (km ²)
50	1.13	0.85
100	1.25	0.71
150	1.37	0.73

The effect of the number of stacks was also investigated. If each stack in the 4-stack arrangement is divided into 4 equal stacks (6,250 lb ea) which are positioned such that their center of charge lies at the position of the original stack while retaining a uniform separation, an arrangement of 16 stacks with half the original separation results. The coalescence and peak overpressure maps produced are shown in Figures 24 and 25. Each map may be compared to the corresponding map in Figures 22 or 23. A significant advantage to smaller subdivisions is indicated. The effect of separation on the maximum effective explosive weight and the blast hazard areas is given in Table 4. Comparison with Table 3 shows a reduction in the predicted blast hazard area but an increase in the regulatory area. The minimum predicted hazard area occurs near the 50-m separation.

Table 4. Variation of Maximum Effective Explosive Weight and Blast Hazard Areas with Separation for Sixteen 6,250-lb Stacks

Separation (m)	Maximum Effective Explosive Weight (lb)	Blast Hazard Area	
		AR 385-64 (km ²)	BWACO (km ²)
25	56,250	1.18	0.64
50	25,000	1.37	0.50
75	25,000	1.57	0.56

Simulations of representative storage configurations with 6 and 10 unequal stacks demonstrate the utility of BWACO. In each case, the stacks have TNT equivalent weights between 20,000 and 30,000 lb and are separated from their neighbors by a distance of 100 m.

The six stacks have a total weight of 150,000 lb. The coalescence and peak overpressure maps are shown in Figure 26. The weight contours in the coalescence map are hard to distinguish but their levels are listed in the legend. The highest weight effective within the region of significant pressure is 99,000 lb. This is significantly less than the total weight. The level of the outermost contour shown on the peak overpressure map is 1.1 psi. This corresponds to the regulatory requirement for the 150,000-lb total weight. The blast hazard area predicted by BWACO using this pressure level is less than half that specified by the regulation. If the larger peak overpressure level corresponding to the maximum effective weight (99,000 lb) were used, an even more favorable comparison would result.

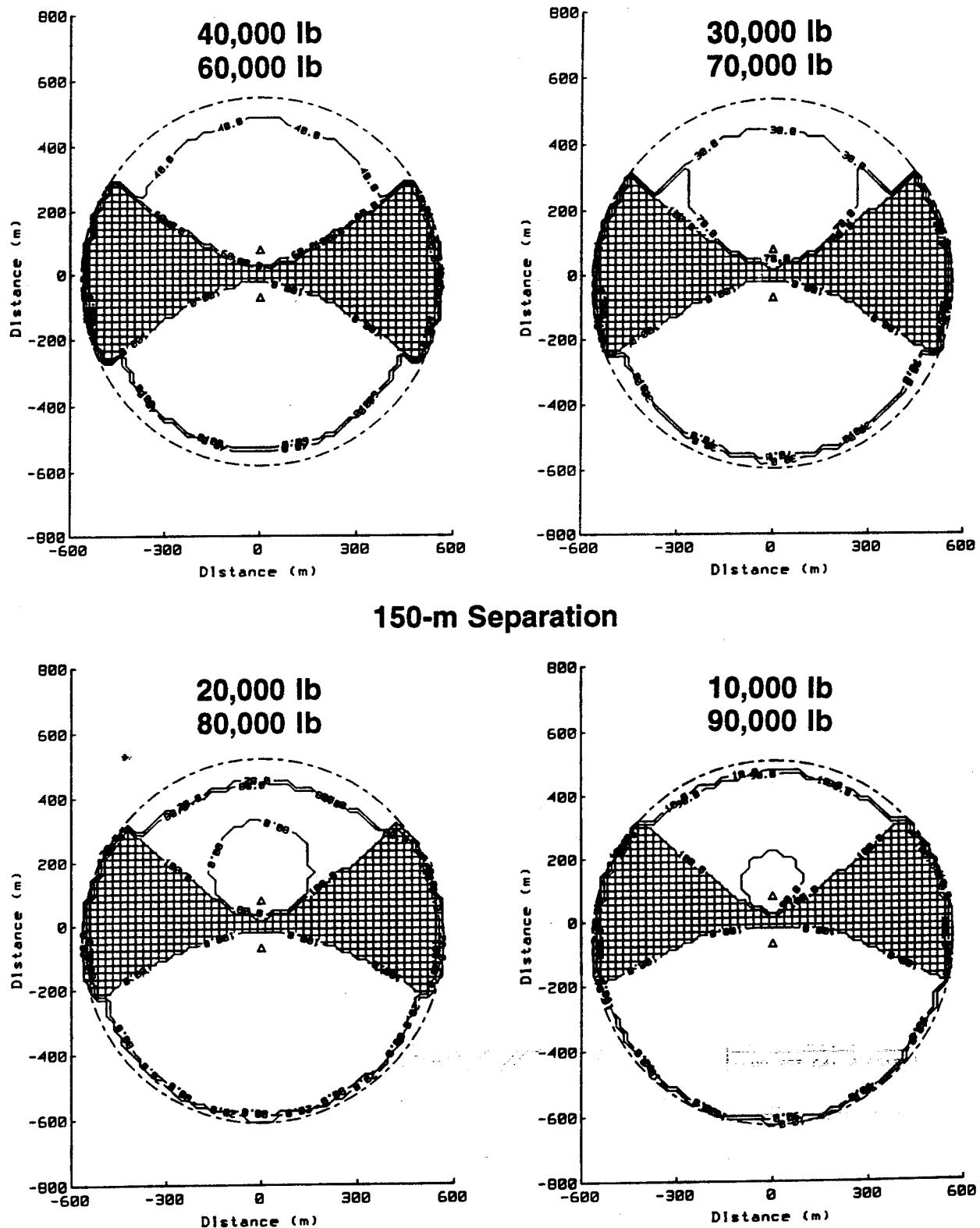
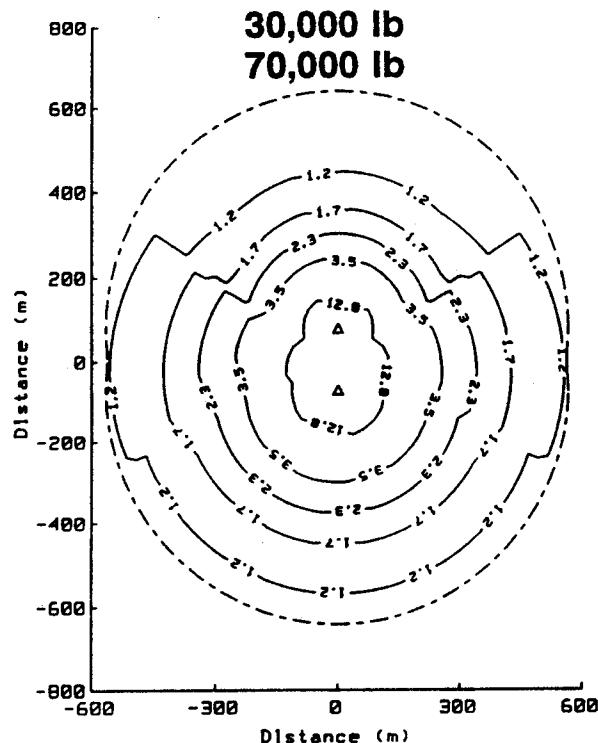
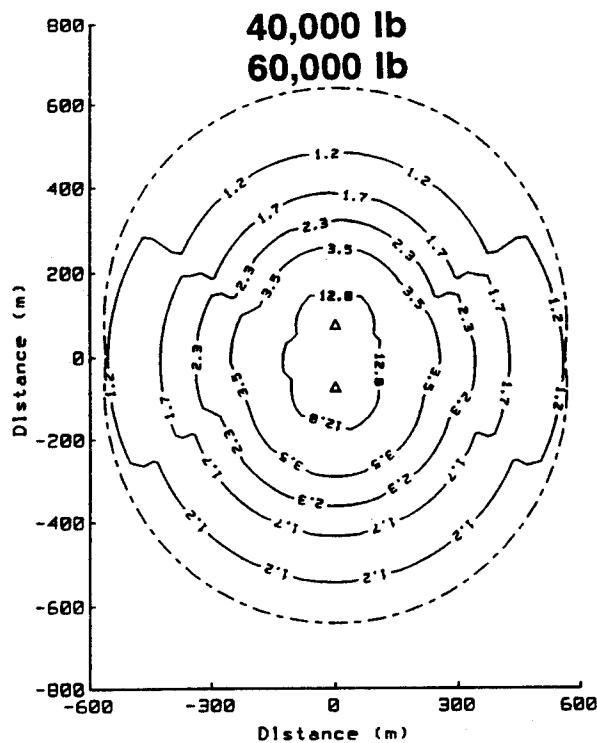


Figure 20. Coalescence maps for large-scale simulations with two unequal stacks having a total weight of 100,000 lb.



150-m Separation

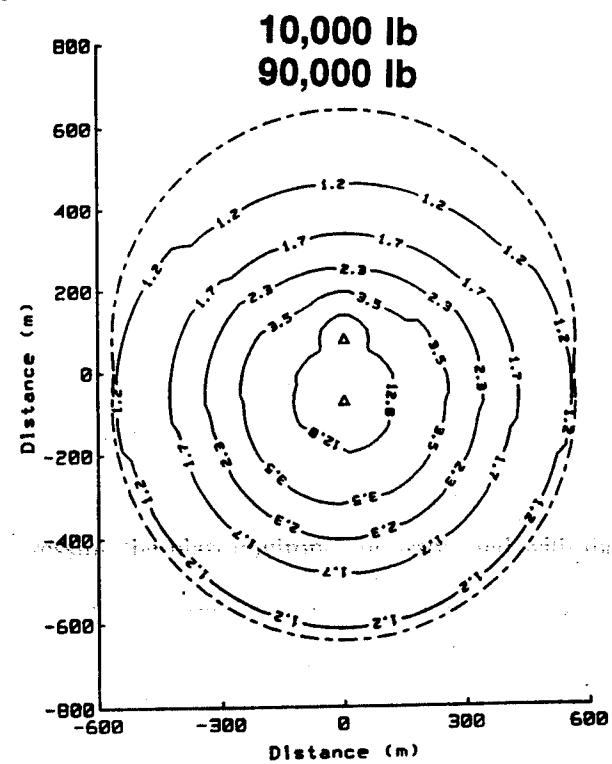
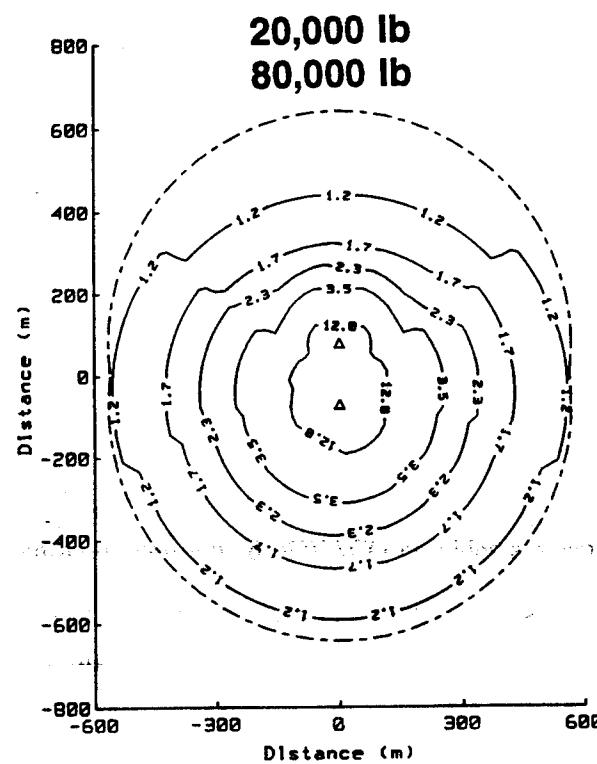


Figure 21. Peak overpressure maps for large-scale simulations with two unequal stacks having a total weight of 100,000 lb.

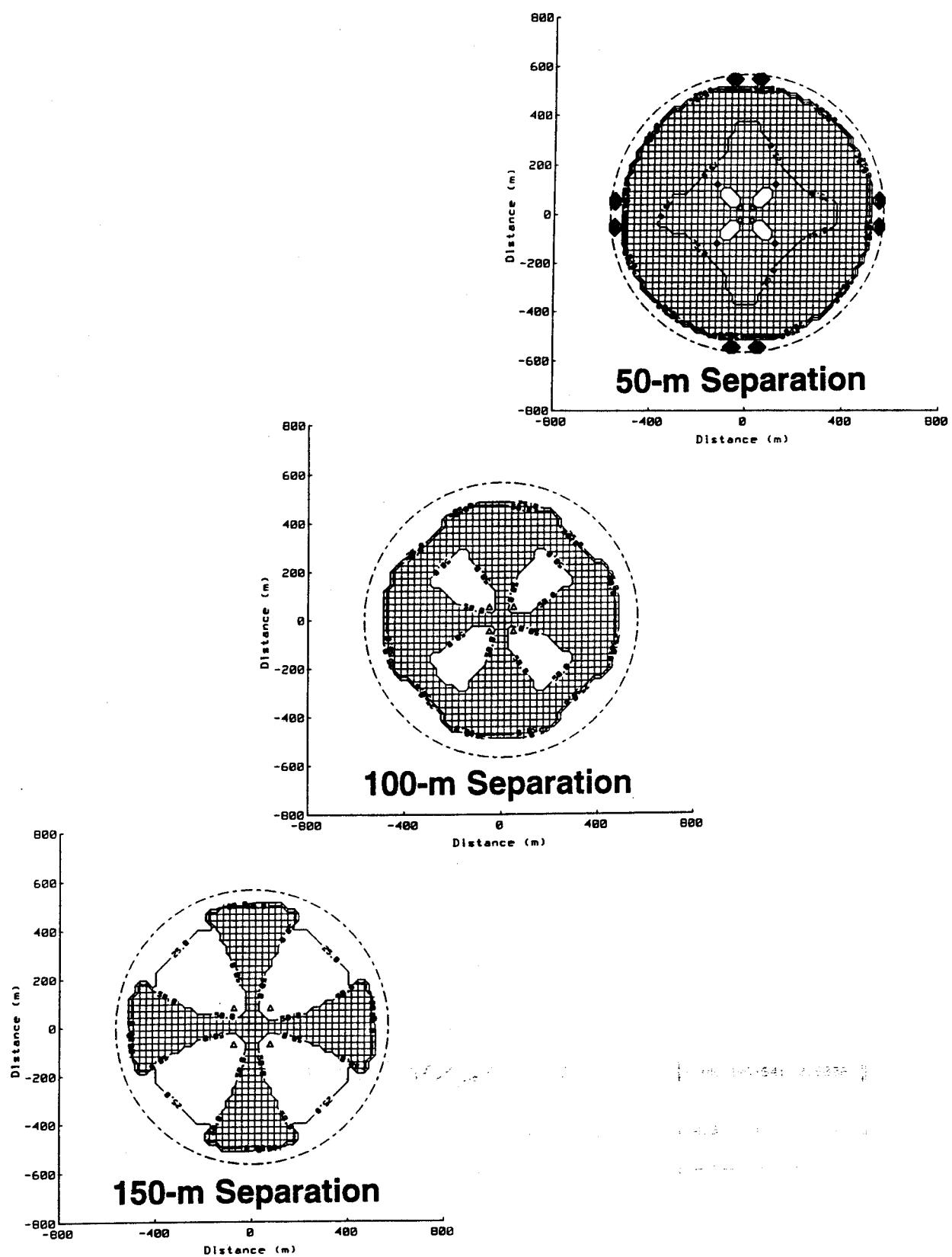


Figure 22. Coalescence maps for large-scale simulations with four 25,000-lb stacks.

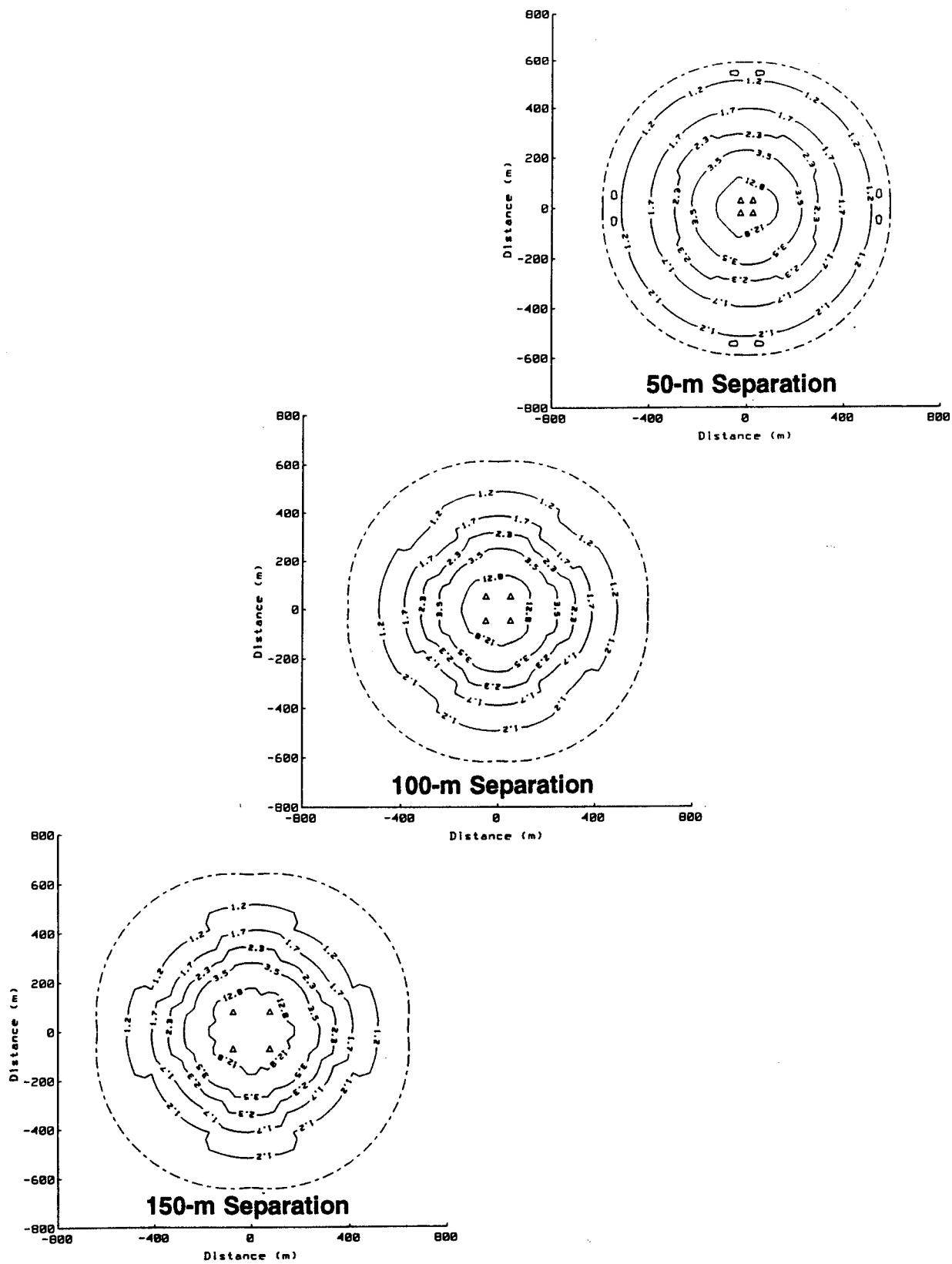


Figure 23. Peak overpressure maps for large-scale simulations with four 25,000-lb stacks.

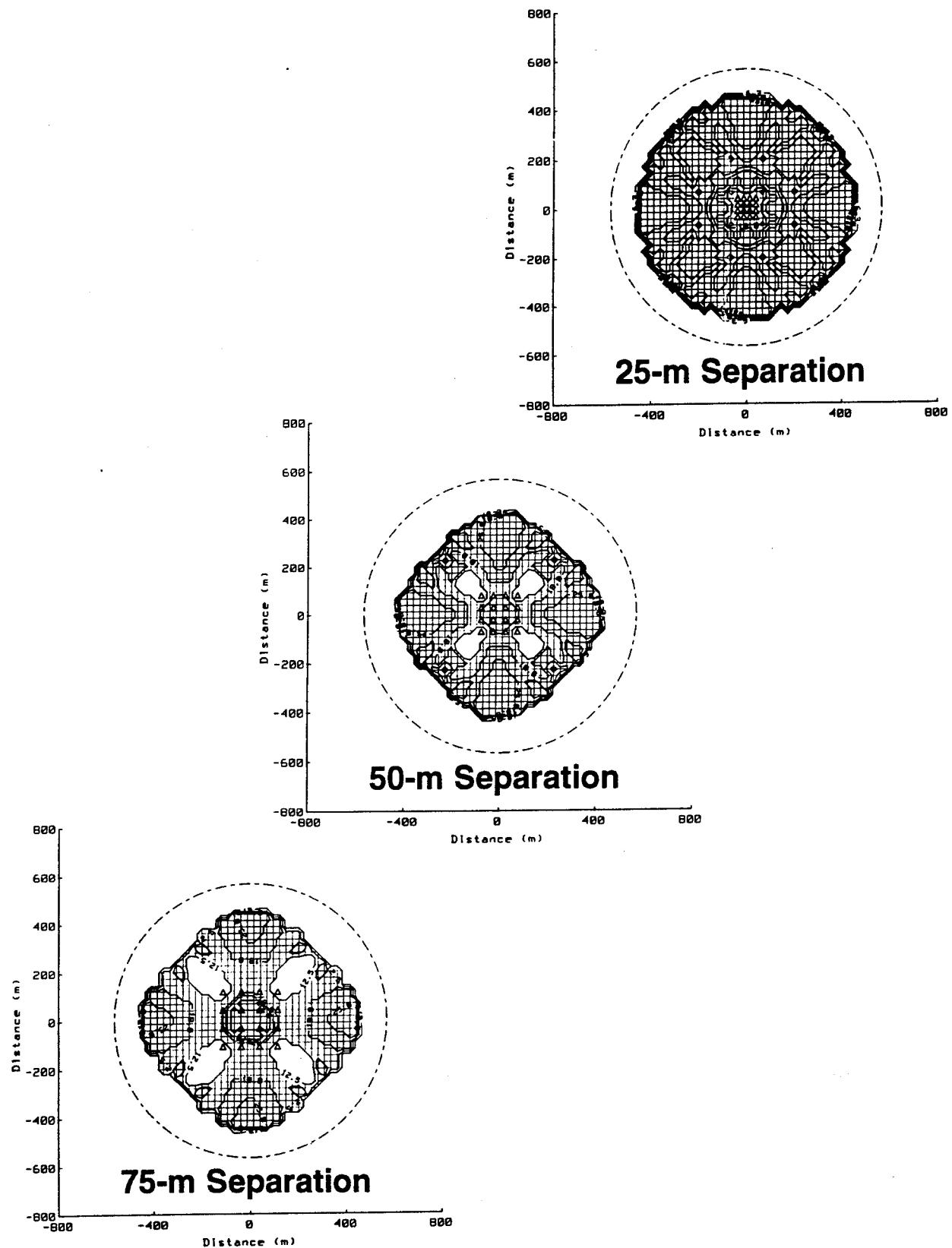


Figure 24. Coalescence maps for large-scale simulations with sixteen 6,250-lb stacks.

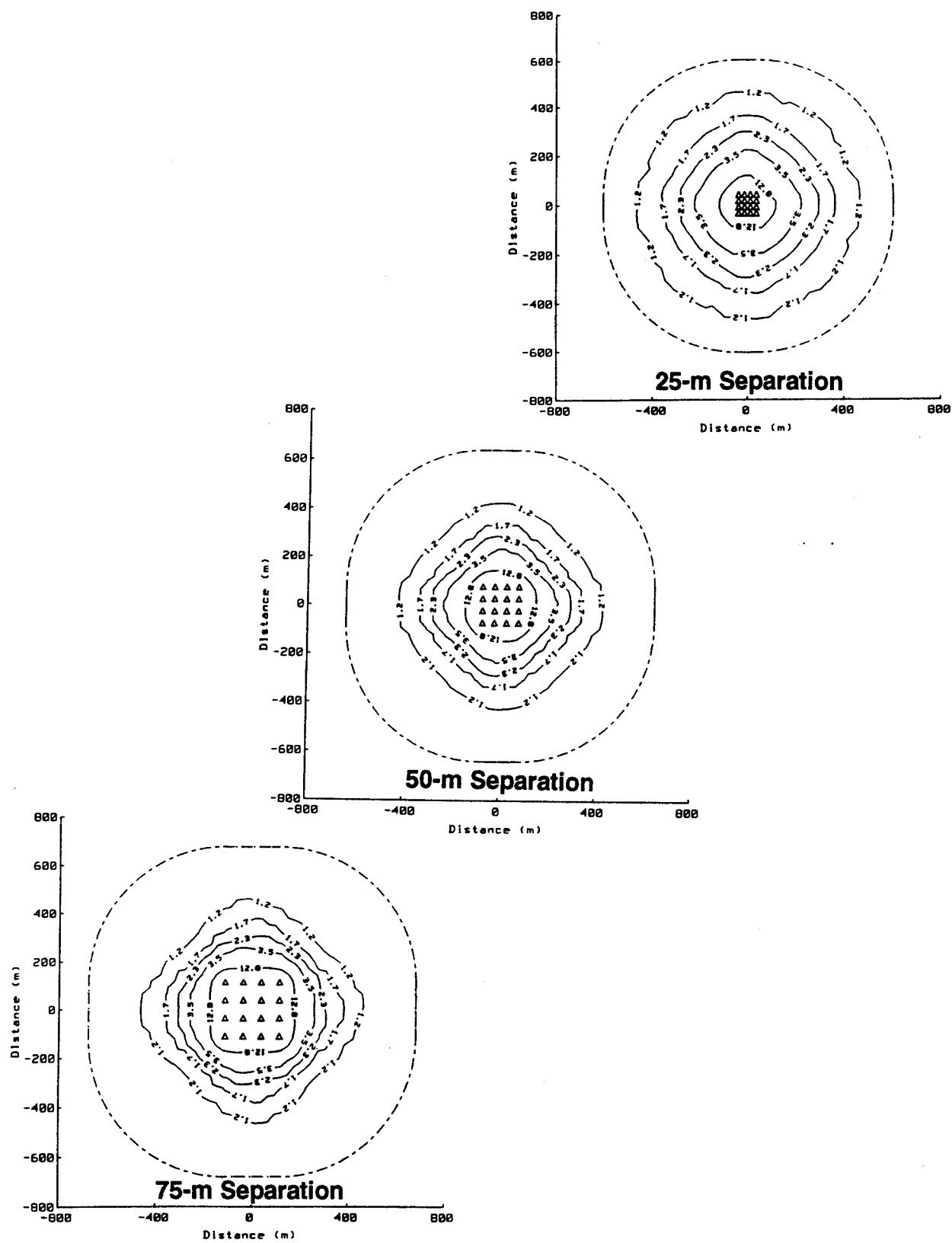


Figure 25. Peak overpressure maps for large-scale simulations with sixteen 6,250-lb stacks.

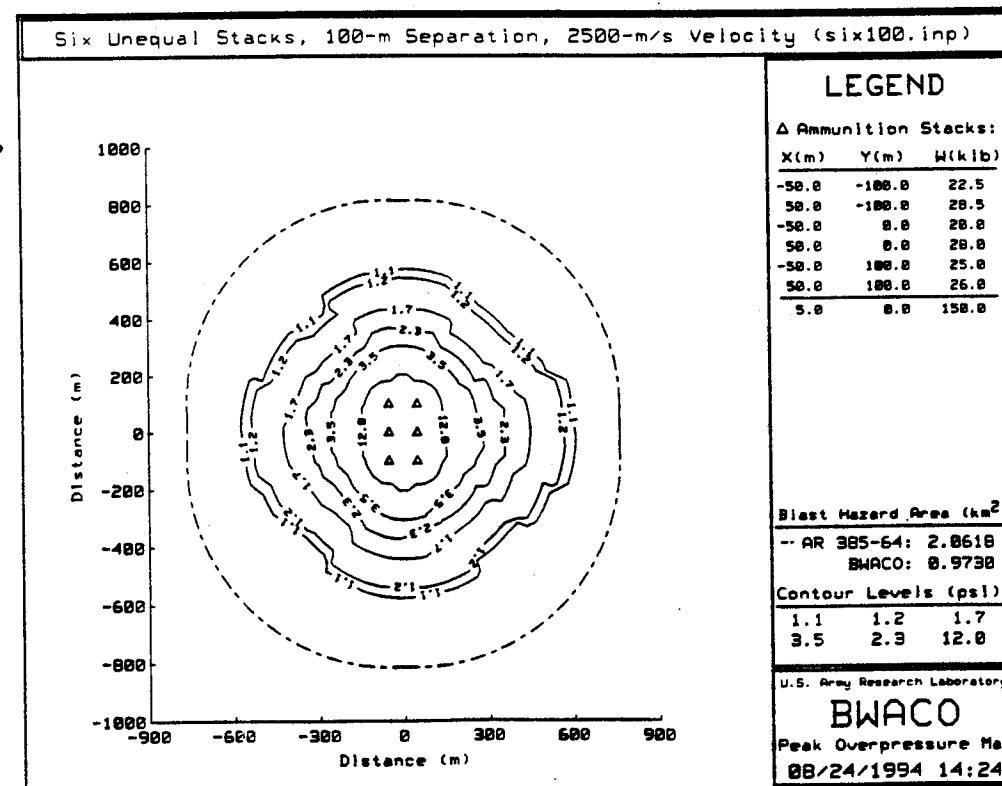
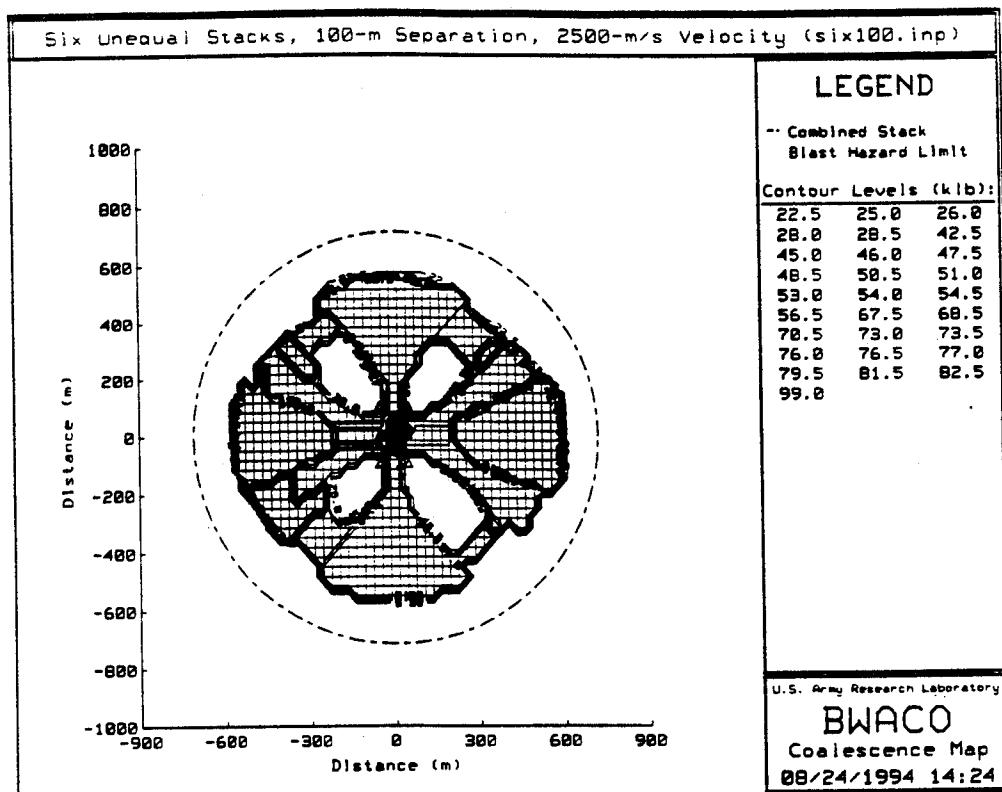


Figure 26. Coalescence and peak overpressure maps for a large-scale simulation with six unequal stacks having a total weight of 150,000 lb.

The ten stacks have a total weight of 250,000 lb, requiring consideration of peak overpressures down to 0.9 psi. The coalescence and peak overpressure maps are shown in Figure 27.

The highest weight effective within the region of significant pressure is 134,000 lb. The ratio of the BWACO hazard area to the regulatory hazard area is even smaller than in the six-stack case. Again, this comparison would become still more favorable if the larger overpressure corresponding to the maximum effective weight were used.

8. SUMMARY AND CONCLUSIONS

In this report, we have explained the assumptions underlying the BWACO algorithms, documented the evolution of BWACO based on comparisons with available experimental data, and demonstrated the application of BWACO to typical large-scale ammunition storage configurations.

Assumptions were made in order to determine the order and timing of the detonation of the stacks following the detonation of any initial donor stack, to establish a criterion for detecting coalescence, and to determine the combined pressure associated with a number of coalesced waves at a point.

Comparison of preliminary results with experimental data obtained by Zaker led to replacement of the standards initially used for the description of blast waves. Use of the 1-KT Standard was found to produce predictions which tended to miss detection of coalescence where the experiments showed that coalescence occurred. It was replaced with a standard based on experimental data reported by Kingery. In addition, a pressure combination algorithm based on Zaker's observation was found to produce more satisfactory results than other algorithms.

BWACO has been adapted for the personal computer with enhanced graphical representations. As currently configured, BWACO provides a means of assessing the blast environment associated with the sequential detonation of an arbitrary arrangement of ammunition stacks. The limitations imposed by the assumptions have not been assessed in realistic configurations.

Applications to a number of problems representative of typical ammunition storage configurations were detailed. The results indicate that regions of significant pressure associated with the coalescence of blast waves from distributed ammunition stacks may be less extensive than corresponding regions associated

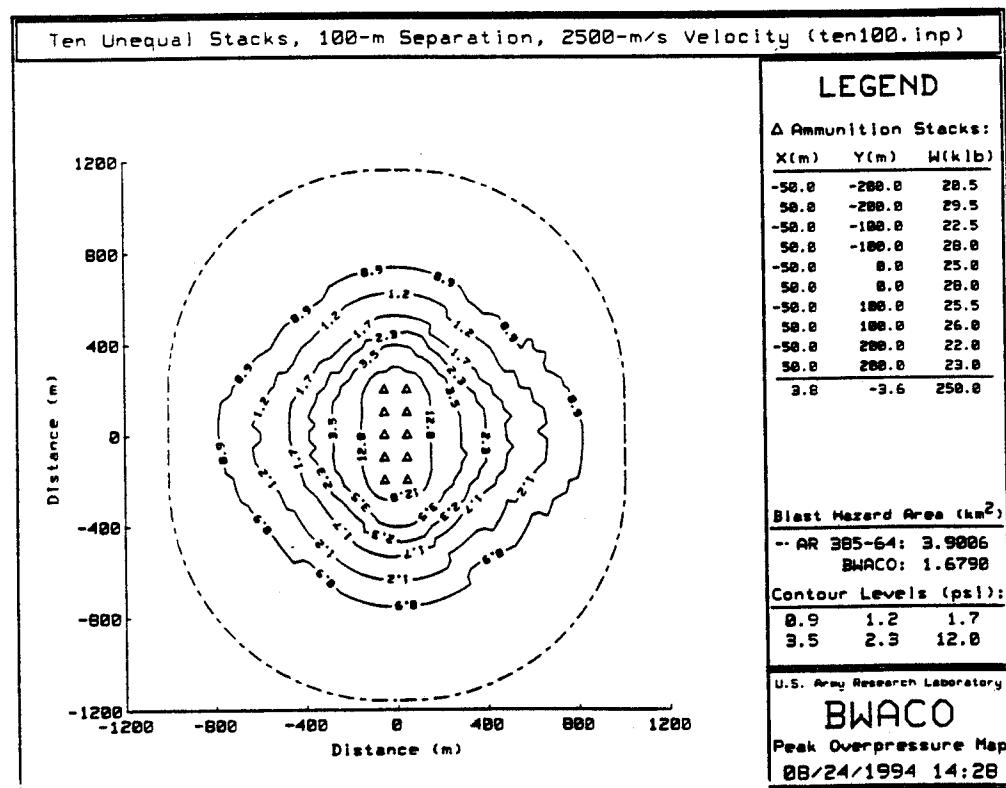
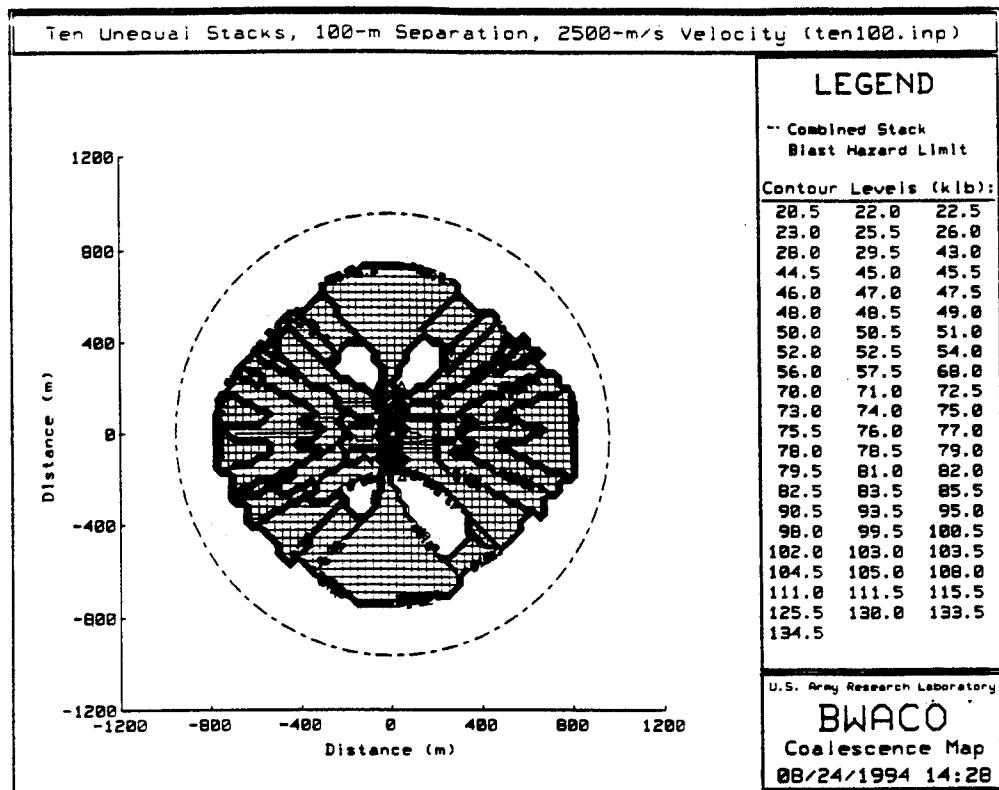


Figure 27. Coalescence and peak overpressure maps for a large-scale simulation with ten unequal stacks having a total weight of 250,000 lb.

with regulatory requirements. An advantage associated with the distribution of ammunition into smaller subdivisions was also demonstrated.

If it proves advantageous, opportunities to apply more sophisticated models can be exploited. Areas where improvement is possible include the propagation model and the pressure combination algorithm, as well as the addition of terrain effects and fragment hazard predictions.

INTENTIONALLY LEFT BLANK.

9. REFERENCES

Brode, H. L. "Quick Estimates of Peak Overpressure From Two Simultaneous Blast Waves." R&D Associates, RDA-TR-107006-008, Marina Del Rey, CA, December 1977.

Department of the Army. AR 385-64, "Ammunition and Explosives Safety Standards." 22 May 1987.

Hikida, S., and C. E. Needham. "Low Altitude Multiple Burst (LAMB) Model." S-Cubed Final Report S-CUBED-R-81-5067, 30 June 1981.

Kingery, C. N., J. H. Keefer, and J. D. Day. "Surface Air Blast Measurements From a 100-Ton TNT Detonation." BRL Memorandum Report No. 1410, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, June 1962.

Kingery, C. N., and B. F. Pannill. "Peak Overpressure vs. Scaled Distance for TNT Surface Bursts (Hemispherical Charges)." BRL Memorandum Report No. 1518, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, April 1964.

Kingery, C. N. "Air Blast Parameters vs. Distance for Hemispherical TNT Surface Bursts." BRL Report No. 1344, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, September 1966.

Needham, C. E., and J. E. Crepeau. "The DNA Nuclear Blast Standard (1 KT)." Systems, Science and Software Report, SSS-R-81-4845, 30 January 1981.

Zaker, T. A. "Far Field Overpressure From Closely Spaced Sequential Detonations." Minutes of the Eleventh Explosive Safety Seminar, 9-10 September 1969.

INTENTIONALLY LEFT BLANK.

APPENDIX A:
DETAILED DESCRIPTION OF BWACO SOURCE CODE

INTENTIONALLY LEFT BLANK.

BACKGROUND

The BWACO program was written to predict regions of blast wave coalescence associated with the sequential mass detonation of distributed ammunition stacks positioned as specified by the user. It accomplishes this by assuming that blast waves propagate independently of one another and that coalescence occurs wherever and whenever the shock associated with one blast wave encroaches into the positive overpressure phase of another blast wave. It further assumes that communication of mass detonation between stacks is controlled by a user-supplied propagation velocity (which might represent the velocity of fragments from the donor stack). It uses this propagation velocity to determine the order and timing of detonation of the stacks. It recognizes that the identity of the initial donor is not generally known and, unless a single donor has been specified by the user, considers all possible donors in order to predict the worst-case environment at any point. It reports the peak overpressure and effective explosive weight contributing to this environment throughout the region of interest.

Although the vast majority of BWACO represents new code, a few of its FORTRAN subroutines and functions were adapted from the DNA Nuclear Blast Standard (1 KT) (hereinafter referred to as the 1-KT Standard), which describes the blast environment produced by a 1-KT nuclear explosion, and from the Low Altitude Multiple Burst (LAMB) model, which describes the blast environment produced by multiple nuclear explosions of arbitrary yield. As a result, much of the terminology used to discuss the program comes from the nuclear blast vocabulary. For example, the net equivalent explosive weight in an ammunition stack is referred to as its yield and detonation of a stack is called a burst.

The organization of BWACO's subroutines and functions is diagrammed in Figure A-1 (where remaining adaptations from the 1-KT Standard or LAMB are rendered white on black). The routines are divided into six modules. The interface module directs BWACO's major activities, configuring the user session, setting the constants, and interfacing with the user by providing a menu of options. These include problem setup and execution, plotting, and reconfiguration of the session. The execution module reads the user's input, determines the region of interest, distributes stations within it, and executes the computation. The plotting module interfaces with the user to produce plots on request. The propagation module determines the order and timing of detonation propagation through the stacks. The coalescence module determines the waveform variables and applies the coalescence criterion at each station. The waveform module computes the required scale factors and determines the required waveform parameters from the applicable standard.

BWACO SUBROUTINES

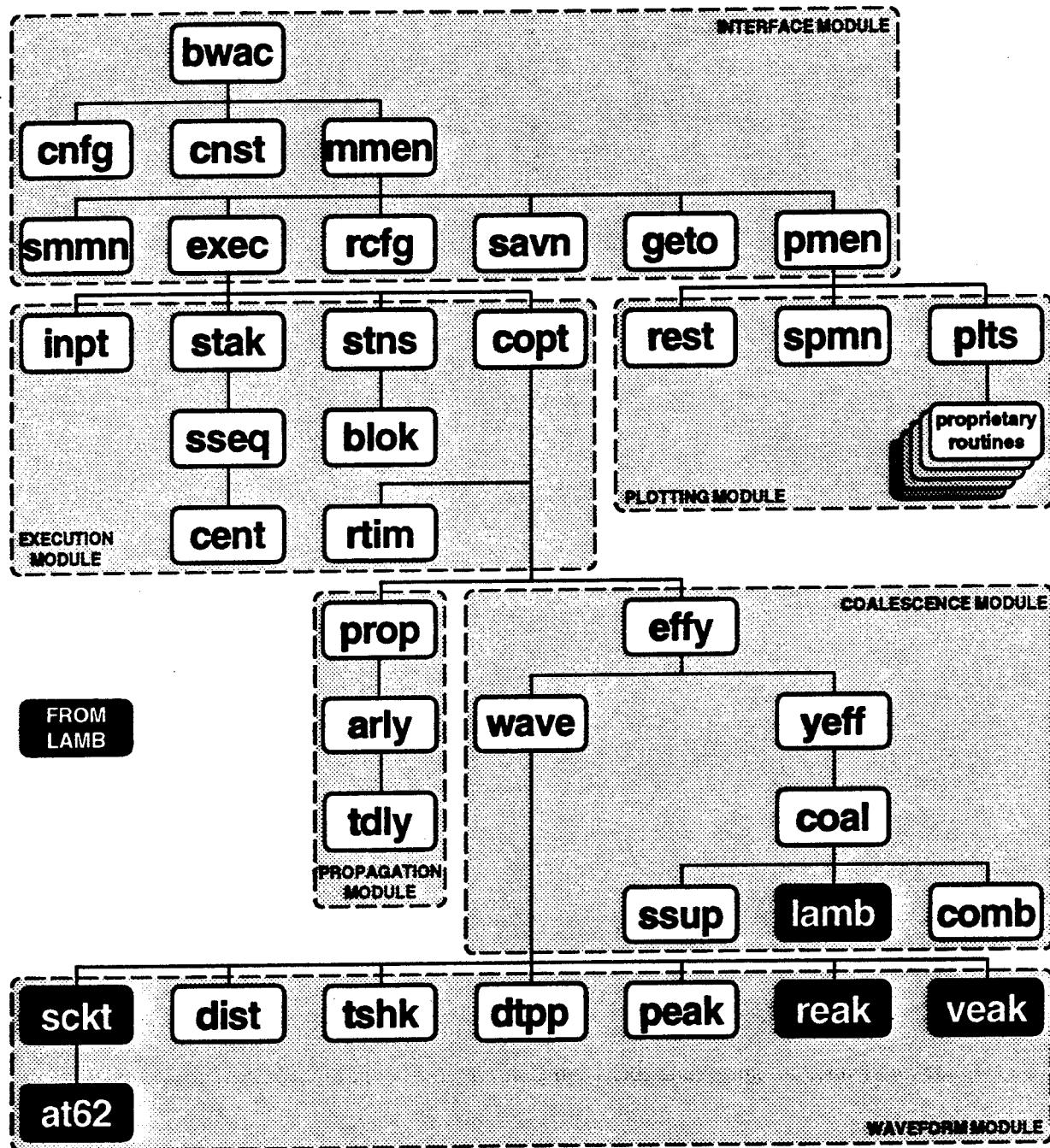


Figure A-1. Schematic of BWACO subroutine calling relationships.

The following discussion details most of the routines, describing the program from the inside out. BWACO's logic is difficult to understand and explain and the inside-out approach facilitates exposition of its operation.

WAVEFORM MODULE

These routines use the applicable blast standard to describe the shock wave, the boundary between the positive and negative phases of the blast wave (point of zero overpressure) and the peak values of overpressure, density, and particle velocity associated with a burst.

Subroutine *at62*, adapted from the 1-KT Standard, is shown in Figure A-2. It computes ambient pressure (Pa), sound speed (cm/s), density (g/cm^3) and temperature (K) from the 1962 U.S. standard temperature atmosphere for a given altitude (cm).

Subroutine *sckt*, adapted from LAMB, is shown in Figure A-3. It computes scale factors for time, distance and pressure for use with the 1-KT Standard given the altitude (cm) and yield (kt) of the burst and the altitude (cm) of the station of interest. Unneeded scale factors have been eliminated from this routine.

Function *dist* is shown in Figure A-4. It computes the radius from a stack to a station given their coordinates.

Function *tshk* is shown in Figure A-4. It computes the time of arrival of the shock wave produced by a 1-KT burst given the scale radius from the center of burst and scales it to the yield of the specified burst.

Function *dtpp* is shown in Figure A-4. It computes the positive phase duration of the blast wave produced by a 1-KT burst given the scale radius from the center of burst and scales it to the yield of the specified burst.

Function *peak* is shown in Figure A-5. It computes the peak overpressure of the blast wave associated with a specified burst given the scale radius from the center of burst.

```

c
c subroutine at62(tty,wsp,cs,wsr,wst)
c
c computes standard atmosphere conditions at a given altitude
c (adapted from DNA 1-KT STANDARD)
c
c tabat(1) = r, the gas constant in erg/mole/K
c tabat(2) = radius of the earth in cm
c tabat(3) = acceleration due to gravity at sea level in cm/s/s
c tabat(4) = molecular weight of air at sea level
c
c tabz = altitude in cm
c tabt = molecular scale temperature in K
c tabl = molecular scale temperature gradient in K/cm
c tabp = pressure in dyne/cm**2
c
c nz = the number of altitudes
c
c dimension tabat(4),tabz(22),tabl(21),tabt(22),tabp(22)
c
c data for temperate atmosphere
c data nz/ 21/, rhoz/ 1.22500000e-03/, tabat/
c + 8.3144000e+07, 6.367488e+08, 9.80665000e+02, 2.8964400e+01/
c data tabz/
c + 0, 1.10190000e+06, 2.00630000e+06, 3.21620000e+06,
c + 4.73500000e+06, 5.24290000e+06, 6.15910000e+06, 7.99940000e+06,
c + 9.00000000e+06, 1.00000000e+07, 1.10000000e+07, 1.20000000e+07,
c + 1.50000000e+07, 1.60000000e+07, 1.70000000e+07, 1.90000000e+07,
c + 2.20000000e+07, 3.00000000e+07, 4.00000000e+07, 5.00000000e+07,
c + 6.00000000e+07, 7.00000000e+07/
c data tabl/
c +-6.49291769e-05, 9.28018576e-08, 9.86255062e-06, 2.77080373e-05,
c +-1.72248474e-07, -1.96000240e-05, -3.91696897e-05, 1.60821507e-07,
c + 2.98166740e-05, 5.02020080e-05, 9.97762300e-05, 2.00108809e-04,
c + 1.49589024e-04, 1.00407490e-04, 6.97598500e-05, 6.68801467e-05,
c + 3.49035000e-05, 3.31099360e-05, 2.58868500e-05, 1.71252960e-05,
c + 1.09162420e-05/
c data tabt/
c + 2.88150000e+02, 2.16604540e+02, 2.16688470e+02, 2.28621170e+02,
c + 2.70704137e+02, 2.70616652e+02, 2.52659110e+02, 1.80575130e+02,
c + 1.80736048e+02, 2.10552722e+02, 2.60754730e+02, 3.60530960e+02,
c + 9.60857386e+02, 1.11044641e+03, 1.21085390e+03, 1.35037360e+03,
c + 1.55101404e+03, 1.83024204e+03, 2.16134140e+03, 2.42020990e+03,
c + 2.59146286e+03, 2.70062528e+03/
c data tabp/
c + 1.01325000e+06, 2.26320000e+05, 5.47486994e+04, 8.68013979e+03,
c + 1.10904998e+03, 5.90004987e+02, 1.82098959e+02, 1.03769924e+01,
c + 1.64379881e+00, 3.00749781e-01, 7.35439451e-02, 2.52169805e-02,
c + 5.06169601e-03, 3.69429709e-03, 2.79259780e-03, 1.68519867e-03,
c + 8.67381898e-04, 1.94317430e-04, 4.15743157e-05, 1.13023464e-05,
c + 3.55894454e-06, 1.22936355e-06/
c data cons/1.389779604e14/
c
c find the table index corresponding to the given altitude
c if(tty.gt.0.0) then
c   do 110 jat=1,nz
c     if(tty-tabz(jat)) 120,130,110
110  continue
c   jat=nz+1
120  jat=jat-1
c   jat=max0(1,jat)
c else
c   jat=1
c endif
c
c compute the standard atmosphere conditions
130 dum2=(tabz(jat)-tty)/(tabat(2)+tty)/(tabat(2)+tabz(jat))
dum3=(tabat(2)+tabz(jat))/(tabat(2)+tty)
var1=tabat(2)*tabl(jat)-tabt(jat)+tabl(jat)*tabz(jat)
var2=(tabt(jat)-tabl(jat)*tabz(jat)+tabl(jat)*tty)/tabt(jat)
fs=-cons*(dum2/var1+tabl(jat)/(var1*var1)*alog(dum3*var2))
wsp=tabp(jat)*exp(fs)
wst=tabt(jat)+tabl(jat)*(tty-tabz(jat))
wsr=wsp*tabt(1)*rhoz/(wst*tabp(1))
cs=sqrt(1.4*wsp/wsr)
c
c return
c
c end

```

Figure A-2. Subroutine *at62* from the waveform module.

```

c_____ S000010
c_____ S000020
c
c subroutine sckt(zbur,wbur,zsta, tscal,rscal,pscal) S000030
c
c computes scale factors for use with 1-kt standard S000040
c (adapted from DNA 1-KT STANDARD) S000050
c
c independent variables: S000060
c zbur = burst altitude in cm above sea level S000070
c zsta = station altitude in cm above sea level S000080
c wbur = yield of the burst in kt S000090
c
c dependent variables: S000100
c tscal = scales actual times to 1-kt sea level S000110
c rscal = scales 1 kt sea level dimensions to actual S000120
c pscal = scales 1 kt sea level pressures to actual S000130
c
c common/conf/ thrd, tpi S000140
c common/cona/ p1,c1,r1,t1 S000150
c
c call at62 (zsta, p3,c3,r3,t3) S000160
c pscal=p3/p1 S000170
c if((zsta-zbur).ge.100.) then S000180
c   call at62(zbur, p2,c2,r2,t2) S000190
c   dz=zsta-zbur S000200
c   b=alog(p3/p2)/dz S000210
c   bzsta=b*zsta S000220
c   a=p3*exp(-bzsta) S000230
c   p3=a*(exp(bzsta)-exp(b*zbur))/(b*dz) S000240
c   b=alog(r3/r2)/dz S000250
c   bzsta=b*zsta S000260
c   a=r3*exp(-bzsta) S000270
c   r3=a*(exp(bzsta)-exp(b*zbur))/(b*dz) S000280
c   c3=sqrt(1.4*p3/r3) S000290
c endif S000300
c   rscal=(wbur*p1/p3)**thrd S000310
c   tscal=c3/(rscal*c1) S000320
c
c   return S000330
c
c   end S000340
c
c_____ S000350
c_____ S000360
c_____ S000370
c_____ S000380
c_____ S000390
c_____ S000400
c_____ S000410
c_____ S000420

```

Figure A-3. Subroutine sckt from the waveform module.

```

c   function dist(xstan,ystan,zstan,xstck,ystck,zstck)          D000010
c   computes the radius from a stack to a station                  D000020
c   independent variables:                                         D000030
c     xstan = x-coordinate of the station                          D000040
c     ystan = y-coordinate of the station                          D000050
c     zstan = z-coordinate of the station                          D000060
c     xstck = x-coordinate of the stack                           D000070
c     ystck = y-coordinate of the stack                           D000080
c     zstck = z-coordinate of the stack                           D000090
c   local variables:                                              D000100
c     deltx = x-component of separation between station and stack D000110
c     delty = y-component of separation between station and stack D000120
c     deltz = z-component of separation between station and stack D000130
c     dist=sqrt(deltx*deltx+delty*delty+deltz*deltz)           D000140
c   return
c   end
c
c   function tshk(tscal,rs1kt)                                     T000010
c   computes the shock arrival time for a                         T000020
c   specified burst as a function of radius                      T000030
c   independent variables:                                         T000040
c     tscal = time scale factor                                T000050
c     rs1kt = 1 kiloton scale radius                          T000060
c   dependent variable:                                         T000070
c     tshk = shock arrival time                                T000080
c   local variable:                                              T000090
c     ts1kt = 1 kiloton scale arrival time                   T000100
c   common/cons/ r0s,c0s,d0s                                     T000110
c   compute the 1-kt shock arrival time                         T000120
c   ts1kt=(rs1kt-d0s*tanh(rs1kt/r0s))/c0s                   T000130
c   scale the arrival time to actual                          T000140
c   tshk=ts1kt/tscal                                         T000150
c   return
c   end
c
c   function dtpp(tscal,rs1kt)                                     D000010
c   computes the positive phase duration for                  D000020
c   a specified burst as a function of radius                  D000030
c   independent variables:                                         D000040
c     tscal = time scale factor                                D000050
c     rs1kt = 1-kiloton scale radius                          D000060
c   dependent variable:                                         D000070
c     dtpp = positive phase duration                         D000080
c   local variable:                                              D000090
c     dt1kt = 1-kiloton scale positive phase duration       D000100
c   common/comp/ a0p,b0p2,c0p,alp,blp,r0p                   D000110
c   apply the lower limit to the radius                      D000120
c   r1=amax1(rs1kt,r0p)                                     D000130
c   compute the 1-kt positive phase duration                D000140
c   dt1kt=c0p*(1.0-alp/r1**blp)*sqrt(alog(a0p*sqrt(r1*r1+b0p2))) D000150
c   scale the positive phase duration to actual            D000160
c   dtpp=dt1kt/tscal                                         D000170
c   return
c   end

```

Figure A-4. Functions *dist*, *tshk*, and *dtpp* from the waveform module.

```

c          function peak(pscal,rs1lb)          P000010
c          computes the peak overpressure as a function P000020
c          of scaled radius using the kingery fit P000030
c          independent variables: P000040
c          pscal = pressure scale factor P000050
c          rs1lb = 1-pound scale radius P000060
c          dependent variable: P000070
c          peak = peak overpressure P000080
c          common/conv/ okt,ocm,odsc,ocmkt3 P000090
c          common/conk/ ak0,ak(8) P000100
c          compute the peak overpressure in psi P000110
c          alrs1=alog(rs1lb) P000120
c          ping=ak0 P000130
c          do 110 i=1,8 P000140
c          ping=ping+ak(i)*alrs1**i P000150
110  continue P000160
c          ping=exp(ping) P000170
c          convert to dyne/cm**2 and scale to sea level P000180
c          peak=odsc*pscal*ping P000190
c          return P000200
c          end P000210
c
c          function reak(pratio,rambi)          R000010
c          computes peak density (adapted from LAMB) R000020
c          independent variables: R000030
c          pratio = pressure ratio R000040
c          rambi = ambient density R000050
c          dependent variable R000060
c          reak = peak density: R000070
c          common/coni/ gamma,gmone,gpone R000080
c          reak=rambi*((gpone*pratio+gmone)/(gmone*pratio+gpone)) R000090
c          return R000100
c          end R000110
c
c          function veak(ppeak,rpeak,rambi)          V000010
c          computes the peak particle velocity (adapted from LAMB) V000020
c          independent variables: V000030
c          rambi = ambient density V000040
c          ppeak = peak overpressure V000050
c          rpeak = peak overdensity V000060
c          dependent variable: V000070
c          veak = peak particle velocity V000080
c          opeak=rpeak-rambi V000090
c          veak=sqrt(ppeak*opeak/(rambi*(rambi+opeak))) V000100
c          return V000110
c          end V000120
c

```

Figure A-5. Functions *peak*, *reak*, and *veak* from the waveform module.

Function *reak*, adapted from LAMB, is shown in Figure A-5. It computes the peak density of the blast wave associated with a specified burst given the pressure ratio and the ambient density. It is only used in conjunction with the LAMB peak overpressure combination algorithm.

Function *veak*, adapted from LAMB, is shown in Figure A-5. It computes the peak particle velocity of the blast wave associated with a specified burst given the peak overpressure and overdensity, and the ambient density. It is only used in conjunction with the LAMB peak overpressure combination algorithm.

COALESCENCE MODULE

These routines determine the effective yield at each station for a specified initial donor stack. It is recognized that more than one group of coalesced waves may pass a station and the yield selected is that associated with the group that produces the largest combined peak overpressure.

Subroutine *wave* is shown in Figure A-6. It cycles through the stacks, computes the required scale factors with calls to *sckt*, computes the stack-to-station radii with calls to *dist*, and computes the waveform variables with calls to *tshk*, *dtpp*, and *peak*, as well as to *reak* and *veak* when the LAMB model is specified.

Subroutine *ssup* is shown in Figure A-7. It totals the yields and, using simple superposition, combines the peak overpressures associated with coalesced blast waves. This option can only be selected by altering the default selection in the file *bwaco.cfg*.

Subroutine *lamb* is shown in Figure A-7. It totals the yields and, using the LAMB algorithm, combines the peak overpressures associated with coalesced blast waves. This option can only be selected by altering the default selection in the file *bwaco.cfg*.

Subroutine *comb* is shown in Figure A-7. It totals the yields and, using the total yield, computes the peak overpressures associated with coalesced blast waves.

Subroutine *coal* is shown in Figure A-8. It is given the index of a specified stack and the coordinates of the current station. It cycles through all the other stacks and, using the shock and zero-overpressure arrival times for all stacks at the current station, it applies the coalescence criterion that the shock from

```

c subroutine wave(istan,jstan)
c cycles through the stacks and computes the waveform variables
c independent variables:
c   istan = station x-index
c   jstan = station y-index
c configuration variable:
c   icomb = pressure combination algorithm switch
c station variables:
c   xstat = array of x-coordinates for the stations
c   ystat = array of y-coordinates for the stations
c   zstat = z-coordinate of stations
c stack variables:
c   nstak = total number of stacks
c   xstak = array of x-coordinates for the stacks
c   ystak = array of y-coordinates for the stacks
c   zstak = z-coordinate for the stacks
c   wstak = array of yields for the stacks
c   tstak = array of stack initiation times
c   tshok = array of shock arrival times for waves from stacks
c   tzero = array of zero-overpressure arrival times
c           for waves from stacks
c   ppeak = array of peak overpressures for waves from stacks
c   rpeak = array of peak densities for waves from stacks
c   vpeak = array of peak velocities for waves from stacks
c local variables:
c   istak = stack counter
c   tscal = time scale factor
c   rscal = radius scale factor
c   pscal = pressure scale factor
c   rstst = stack to station radius
c   rs1kt = radius scaled to 1 kiloton
c   rs1lb = radius scaled to 1 pound
c   pratio = pressure ratio
c
c common/conv/  okt,ocm,odsc,ocmkt3
c common/icfg/  icomb,iscrnl,iprnt
c common/ambi/  pambi,rambi
c common/sttp/  xstat(120),ystat(120),zstat
c common/stkn/  nstak
c common/stkp/  xstak(100),ystak(100),zstak
c common/stkt/  vstak(100),tstak(100)
c common/stky/  wstak(100)
c common/stkw/  tshok(100),tzero(100),
c +           ppeak(100),rpeak(100),vpeak(100)
c
c cycle through the stacks
c do 110 istak=1,nstak
c
c     compute the scale factors
c     call skct(zstak,wstak(istak),zstat, tscal,rscal,pscal)
c
c     compute the stack to station radius
c     +   rstst=dist(xstat(istan),ystat(istan),zstat,
c                   xstak(istak),ystak(istak),zstak)
c
c     scale the radius to 1-kt
c     rs1kt=rstst/rscal
c
c     compute the shock and zero-overpressure arrival times
c     tshok(istak)=tshk(tscal,rs1kt)+tstak(istak)
c     tzero(istak)=dtpp(tscal,rs1kt)+tshok(istak)
c
c     scale the radius to 1-lb and nondoubled yield
c     rs1lb=ocmkt3*rs1kt
c
c     compute the peak overpressure, density and particle velocity
c     ppeak(istak)=peak(pscal,rs1lb)
c     if(icomb.eq.1) then
c       pratio=(ppeak(istak)+pambi)/pambi
c       rpeak(istak)=reak(pratio,rambi)
c       vpeak(istak)=veak(ppeak(istak),rpeak(istak),rambi)
c       rpeak(istak)=rpeak(istak)-rambi
c     endif
c
c 110 continue
c
c   return
c
c end

```

Figure A-6. Subroutine *wave* from the coalescence module.

```

c
c subroutine ssup(ncoal,wcoal,pcoal, wcomb,pcomb)
c combines yields and peak overpressures using simple superposition
c
c independent variables:
c   ncoal = number of coalesced waves
c   wcoal = array of yields of sources of coalesced waves
c   pcoal = array of peak overpressures from coalesced waves
c
c dependent variables:
c   wcomb = combined yield from sources of all coalesced waves
c   pcomb = combined peak overpressure of all coalesced waves
c
c local variable:
c   iwave = wave counter
c
c dimension wcoal(100),pcoal(100)
c
c wcomb=0.
c pcomb=0.
c do 110 iwave=1,ncoal
c   wcomb=wcomb+wcoal(iwave)
c   pcomb=pcomb+pcoal(iwave)
c 110 continue
c
c return
c
c end
c
c
c subroutine lamb(ncoal,rasta,wcoal,pcoal,rcoal,vcoal, wcomb,pcomb)
c combines yields and peak overpressures using LAMB algorithm
c
c independent variables:
c   ncoal = number of coalesced waves
c   rasta = ambient density at the current station
c   wcoal = array of yields of sources of coalesced waves
c   pcoal = array of peak overpressures from coalesced waves
c   rcoal = array of peak densities from coalesced waves
c   vcoal = array of peak particle velocities from coalesced waves
c
c dependent variables:
c   wcomb = combined yield from sources of all coalesced waves
c   pcomb = combined peak overpressure of all coalesced waves
c
c local variables:
c   iwave = wave counter
c   rcomb = combined peak density of all coalesced waves
c   vcomb = combined peak velocity of all coalesced waves
c   ecomb = combined peak kinetic energy of all coalesced waves
c
c dimension wcoal(100),pcoal(100),rcoal(100),vcoal(100)
c
c wcomb=0
c rcomb=rasta
c pcomb=0
c vcomb=0
c ecomb=0
c do 110 iwave=1,ncoal
c   wcomb=wcomb+wcoal(iwave)
c   rcomb=rcomb+rcoal(iwave)
c   rv=(rasta+rcoal(iwave))*vcoal(iwave)
c   vcomb=vcomb+rv
c   ecomb=ecomb+rv*vcoal(iwave)
c   pcomb=pcomb+pcoal(iwave)
c 110 continue
c   vcomb=vcomb/rcomb
c   pcomb=ecomb+0.6*(ecomb-rcomb*vcomb*vcomb)
c
c return
c
c end
c
c
c subroutine comb(ncoal,xcoal,ycoal,wcoal,xstat,ystat,zstat,wcomb,
c + pcomb)
c combines peak overpressures using pressure from combined stacks
c
c independent variables:
c   ncoal = number of coalesced waves
c   xcoal = array of x-coordinates of sources of coalesced waves
c   ycoal = array of y-coordinates of sources of coalesced waves
c   wcoal = array of yields of sources of coalesced waves
c   xstat = x-coordinate of the current station
c   ystat = y-coordinate of the current station
c   zstat = z-coordinate of the current station
c
c dependent variables:
c   wcomb = combined yield from sources of all coalesced waves
c   pcomb = combined peak overpressure of all coalesced waves
c
c stack variable:
c   ztak = z-coordinate for the stacks
c
c local variables:
c   tcsc1 = time scale factor for combined stacks
c   rcscl = radius scale factor for combined stacks
c   pcsc1 = pressure scale factor for combined stacks
c   xcomb = x-coordinate of center of combined stacks
c   ycomb = y-coordinate of center of combined stacks
c   rsllb = 1-lb scale radius
c
c common/conv/ okt,ocm,odsc,ocmkt3
c common/stkp/ xstat(100),ystak(100),ztak
c
c dimension xcoal(100),ycoal(100),wcoal(100)
c
c find weight and center of combined stacks
c call sseq(ncoal,xcoal,ycoal,wcoal, xcomb,ycomb,wcomb)
c
c compute the scale factors
c call scft(ztak,wcomb,zstat, tcsc1,rcscl,pcsc1)
c
c compute the 1-lb scale radius from the center of
c the combined stacks using the nondoubled yield
c rsllb=ocmkt3*dist(xstat,ystat,zstat,xcomb,ycomb,ztak)/rcscl
c
c compute the combined pressure
c pcomb=peak(pcsc1,rsllb)
c
c return
c
c end
c

```

Figure A-7. Subroutines ssup, lamb, and comb from the coalescence module.

```

0000010
0000020
0000030
0000040
0000050
0000060
0000070
0000080
0000090
0000100
0000110
0000120
0000130
0000140
0000150
0000160
0000170
0000180
0000190
0000200
0000210
0000220
0000230
0000240
0000250
0000260
0000270
0000280
0000290
0000300
0000310
0000320
0000330
0000340
0000350
0000360
0000370
0000380
0000390
0000400
0000410
0000420
0000430
0000440
0000450
0000460
0000470
0000480
0000490
0000500
0000510
0000520
0000530
0000540
0000550
0000560
0000570
0000580
0000590
0000600
0000610
0000620
0000630
0000640
0000650
0000660
0000670
0000680
0000690
0000700
0000710
0000720
0000730
0000740
0000750
0000760
0000770
0000780
0000790
0000800
0000810
0000820
0000830
0000840
0000850
0000860
0000870
0000880
0000890
0000900
0000910
0000920
0000930
0000940
0000950
0000960
0000970
0000980
0000990
0001000
0001010
0001020
0001030
0001040
0001050
0001060
0001070
0001080
0001090
0001100
0001110
0001120
0001130
0001140
0001150
0001160
0001170
0001180
0001190
0001200
0001210
0001220
0001230
0001240

subroutine coal(ispec,xstat,ystat,zstat,rasta, ncoal,wcomb,pcomb)
  applies the coalescence criterion and determines the
  total yield and combined peak overpressure from all
  waves which coalesce with the wave from a specified stack
  independent variables:
  ispec = index of the specified stack
  xstat = x-coordinate of the current station
  ystat = y-coordinate of the current station
  zstat = z-coordinate of the current station
  rasta = ambient density at the current station
  dependent variables:
  ncoal = number of coalesced waves
  wcomb = combined yield of the sources of all coalesced waves
  pcomb = combined peak overpressure of all coalesced waves
  configuration variable:
  icomb = pressure combination algorithm switch
  stack variables:
  nstak = total number of stacks
  xstak = array of x-coordinates for the stacks
  ystak = array of y-coordinates for the stacks
  zstak = array of z-coordinates for the stacks
  ppeak = array of peak overpressures for waves from the stacks
  rpeak = array of peak densities for waves from the stacks
  vpeak = array of peak velocities for waves from the stacks
  tpeak = array of shock arrival times for waves from the stacks
  tzero = array of zero-overpressure arrival times
  for waves from the stacks
  local variables:
  istak = stack counter
  xcoal = array of x-coordinates of sources of coalesced waves
  ycoal = array of y-coordinates of sources of coalesced waves
  wcoal = array of yields of sources of coalesced waves
  pcoal = array of peak overpressures from coalesced waves
  rcoal = array of peak densities from coalesced waves
  vcoal = array of peak particle velocities from coalesced waves
  common/icfg/ icomb, iscrn, iprint
  common/stkn/ nstak
  common/stkp/ xstak(100),ystak(100),zstak
  common/stky/ wstak(100)
  common/stkw/ tshok(100),tzero(100),
  + ppeak(100),rpeak(100),vpeak(100)
  dimension xcoal(100),ycoal(100),wcoal(100),
  + pcoal(100),rcoal(100),vcoal(100)
  + initialize the number of coalesced waves and coalescence arrays
  ncoal=1
  xcoal(1)=xstak(ispec)
  ycoal(1)=ystak(ispec)
  wcoal(1)=wstak(ispec)
  pcoal(1)=ppeak(ispec)
  if(icomb.eq.1) then
    rcoal(1)=rpeak(ispec)
    vcoal(1)=vpeak(ispec)
  endif
  cycle through the stacks
  do 110 istak=1,nstak
    if the current stack is not the specified stack
    if(istak.ne.ispec) then
      if the wave from the current stack is coalesced
      with the wave from the specified stack
      if(tshok(istak).ge.tshok(ispec)) then
        if(tshok(istak).le.tzero(ispec)) then
          increment the number of coalesced waves
          ncoal=ncoal+1
          store the peak overpressure, overdensity and velocity
          xcoal(ncoal)=xstak(istak)
          ycoal(ncoal)=ystak(istak)
          wcoal(ncoal)=wstak(istak)
          pcoal(ncoal)=ppeak(istak)
          rcoal(ncoal)=rpeak(istak)
          vcoal(ncoal)=vpeak(istak)
          if(icomb.eq.1) then
            rcoal(ncoal)=rpeak(istak)
            vcoal(ncoal)=vpeak(istak)
          endif
        endif
      endif
    endif
  110 continue
  combine the yields and peak
  overpressures of the coalesced waves
  if(ncoal.gt.1) then
    use simple superposition
    if(icomb.eq.0) call ssup(ncoal,wcoal,pcoal, wcomb,pcomb)
    use the LAMB algorithm
    if(icomb.eq.1) call lamb(ncoal,rasta,wcoal,pcoal,rcoal,vcoal,
  + wcomb,pcomb)
    use the combined stacks
    if(icomb.eq.2) call comb(ncoal,xcoal,ycoal,wcoal,xstat,ystat,
  + zstat, wcomb,pcomb)
  else
    use the yield and peak overpressure from the specified stack
    wcomb=wcoal(1)
    pcomb=pcoal(1)
  endif
  return
end

```

Figure A-8. Subroutine *coal* from the coalescence module.

one wave lie in the positive overpressure phase of another wave. In this way, it determines the number of stacks producing waves which coalesce with the wave from the specified stack as well as the total yield and combined peak overpressure (using calls to *ssup*, *lamb*, or *comb* depending on how the session is configured) of the coalesced waves. It returns the number of coalesced waves as well as the associated total yield and combined peak overpressure.

Subroutine *yeff* is shown in Figure A-9. It is given the coordinates of the current station. It cycles through the stacks, calling *coal*, and determines the effective yield associated with the group of coalesced waves producing the largest combined peak overpressure at the current station. It returns the number of coalesced waves in the group, the effective yield and the combined peak overpressure.

Subroutine *effy* is shown in Figure A-10. It cycles through the stations and computes the waveform variables for each stack with calls to *wave* and the effective yields and combined peak overpressures with calls to *yeff*. It saves the peak overpressure and effective yield associated with the largest combined peak overpressure at each station.

PROPAGATION MODULE

These routines determine the order in which the stacks detonate and the time of detonation of each stack for a given initial donor.

Function *tdly* is shown in Figure A-11. It is given the coordinates of a donor and an acceptor stack as well as the propagation velocity of the donor. It computes and returns the initiation delay (communication time) between the donor and acceptor stack.

Subroutine *arly* is shown in Figure A-12. It is given the coordinates, initiation time, and propagation velocity of a donor stack. It cycles through all the other stacks which have not yet become donors and, with calls to *tdly*, determines the initiation time of the earliest acceptor for that donor. It assigns that time to the stack initiation time array if it is earlier than the time already assigned. It returns the index of that acceptor and its initiation time.

```

c _____ Y000010
c _____ Y000020
c subroutine yeff(xstat,ystat,zstat,rambi, icsta,wstat,pstat) Y000030
c _____ Y000040
c computes the effective yield and combined Y000050
c peak overpressure at a current station Y000060
c _____ Y000070
c independent variables: Y000080
c xstat = x-coordinate of the current station Y000090
c ystat = y-coordinate of the current station Y000100
c zstat = z-coordinate of the current station Y000110
c rambi = ambient density at the current station Y000120
c _____ Y000130
c dependent variables: Y000140
c icsta = coalescence status at the current station Y000150
c wstat = effective yield at the current station Y000160
c pstat = combined peak overpressure at the current station Y000170
c _____ Y000180
c stack variable: Y000190
c nstak = total number of stacks Y000200
c _____ Y000210
c local variables: Y000220
c istak = stack counter Y000230
c ncoal = number of waves coalesced with Y000240
c the wave from the current stack Y000250
c wcomb = combined yield at the current station Y000260
c pcomb = combined peak overpressure at the current station Y000270
c _____ Y000280
c common/stkn/ nstak Y000290
c _____ Y000300
c initialize the coalescence status, effective Y000310
c yield and combined peak overpressure Y000320
c icsta=0 Y000330
c wstat=0. Y000340
c pstat=0. Y000350
c _____ Y000360
c cycle through the stacks Y000370
c do 110 istak=1,nstak Y000380
c _____ Y000390
c determine the total yield and combined peak overpressure from Y000400
c all waves which coalesce with the wave from the current stack Y000410
c call coal(istak,xstat,ystat,zstat,rambi, ncoal,wcomb,pcomb) Y000420
c _____ Y000430
c save the coalescence status, total yield Y000440
c and combined peak overpressure associated Y000450
c with the largest combined peak overpressure Y000460
c if(pcomb.gt.pstat) then Y000470
c     icsta=ncoal Y000480
c     wstat=wcomb Y000490
c     pstat=pcomb Y000500
c   endif Y000510
c 110 continue Y000520
c _____ Y000530
c return Y000540
c _____ Y000550
c end Y000560
c _____ Y000570
c _____ Y000580
c _____

```

Figure A-9. Subroutine yeff from the coalescence module.

```

c subroutine effy
c computes the effective yield associated with the
c largest combined peak overpressure at each station
c
c station variables:
c nstat = number of stations
c lblok = length of station blocks
c nblok = number of station blocks
c xstat = array of x-coordinates for the stations
c ystat = array of y-coordinates for the stations
c zstat = z-coordinate of stations
c icoal = array of coalescence statuses for the stations
c wstat = array of effective yields for the stations
c pstat = array of effective peak overpressures for the stations
c rambi = ambient density
c
c local variables:
c istrat = station counter
c istan = station x-index
c jstan = station y-index
c icoal = coalescence status at current station
c wscur = effective yield at current station
c pscur = combined peak overpressure at current station
c
c common/allow/ pallo
c common/ambi/ rambi,rambi
c common/sttn/ nstat
c common/sttb/ lblok,nblok
c common/sttp/ xstat(120),ystat(120),zstat
c common/sttv/ icoal(3600),pstat(3600),wstat(3600)
c
c initialize the station report
c write(6,2010) 0,nstat
c
c cycle through the stations
c do 110 istrat=1,nstat
c
c     report the current station
c     write(6,2020) istrat,nstat
c
c     compute the station indices
c     istan=mod(istrat-1,lblok)+1
c     jstan=(istrat-1)/lblok+1
c
c     compute waveform variables
c     call wave(istan,jstan)
c
c     compute the effective yield and combined
c     peak overpressure at the current station
c     call yeff(xstat(istan),ystat(jstan),zstat,rambi,
c
c     +           icoal(jstan),wscur,pscur)
c
c     save the effective yield and combined peak
c     overpressure associated with the largest
c     combined peak overpressure for any donor
c     if(pscur.gt.pstat(istrat)) then
c         if(pscur.lt.pallo) then
c             icoal(istrat)=0
c             wstat(istrat)=0.
c         else
c             icoal(istrat)=pscur
c             wstat(istrat)=wscur
c         endif
c         pstat(istrat)=pscur
c     endif
c
c 110 continue
c
c     return
c
c     format statements
2010 format(' Computing coalescence at station ',i4,' of ',i4)
2020 format('+ Computing coalescence at station ',i4,' of ',i4)
c
c     end

```

Figure A-10. Subroutine *eff* from the coalescence module.

```

c_____ T000010
c_____ T000020
c_____ function tdly(xacpt,yacpt,xdonr,ydonr,vdonr) T000030
c_____ T000040
c_____ computes the initiation delay between a given donor and acceptor T000050
c_____ T000060
c_____ independent variables: T000070
c_____ xacpt = x-coordinate of the acceptor T000080
c_____ yacpt = y-coordinate of the acceptor T000090
c_____ xdonr = x-coordinate of the donor T000100
c_____ ydonr = y-coordinate of the donor T000110
c_____ vdonr = propagation velocity of the donor T000120
c_____ T000130
c_____ dependent variable: T000140
c_____ tdly = initiation delay T000150
c_____ T000160
c_____ local variables: T000170
c_____ deltx = x-coordinate difference T000180
c_____ delty = y-coordinate difference T000190
c_____ T000200
c_____ if(vdonr.eq.0.0) then T000210
c_____ T000220
c_____ infinite propagation velocity T000230
c_____ tdly=0.0 T000240
c_____ T000250
c_____ else T000260
c_____ T000270
c_____ finite propagation velocity T000280
c_____ deltx=xacpt-xdonr T000290
c_____ delty=yacpt-ydonr T000300
c_____ tdly=sqrt(deltx*deltx+delty*delty)/vdonr T000310
c_____ T000320
c_____ endif T000330
c_____ T000340
c_____ return T000350
c_____ T000360
c_____ end T000370
c_____ T000380
c_____

```

Figure A-11. Function *tdly* from the propagation module.

Figure A-12. Subroutine *arly* from the propagation module.

Subroutine *prop* is shown in Figure A-13. It is given the index of the current initial donor stack. It cycles through all the stacks which have become donors to determine the next donor using calls to *arly*. It repeats the process using that donor and continues until initiation times have been assigned to all of the stacks.

EXECUTION MODULE

These routines read the user's input, set up the requested computation, and execute it.

Subroutine *inpt*, shown in Figure A-14, reads the user's input from the file *bwaco.inp*.

Subroutine *cent*, shown in Figure A-15, computes either coordinate of the center of charge of a given stack arrangement.

Subroutine *sseq*, shown in Figure A-15, calls *cent* to compute both coordinates of the center of charge of a given stack arrangement.

Subroutine *stak*, shown in Figure A-16, calls *sseq* to compute the single-stack equivalent of all the stacks, converts the stack variables to 1-KT Standard units and doubles their yields to simulate the effects of the ground plane.

Subroutine *blok*, shown in Figure A-17, determines the station arrangement that best covers the region of interest.

Subroutine *stns*, shown in Figure A-18, determines the region of interest (if none is specified by the user), determines the station arrangement by calling *blok*, defines the station coordinates, and initializes the station variables.

Subroutine *rtim*, shown in Figure A-19, computes and reports the run time for the coalescence computation.

```

c----- P000010
c subroutine prop(idoni) P000020
c computes initiation times of stacks P000030
c independent variable: P000040
c idoni = index of initial donor P000050
c stack variables: P000060
c nstak = number of stacks P000070
c xstak = array of x-coordinates for the stacks P000080
c ystak = array of y-coordinates for the stacks P000090
c vstak = array of stack propagation velocities P000100
c tstak = array of initiation times for the stacks P000110
c mstak = array of initiation statuses for the stacks P000120
c local variables: P000130
c istak = stack counter P000140
c ierly = index of earliest acceptor P000150
c terly = initiation time of earliest acceptor P000160
c idonx = index of next donor P000170
c tnext = earliest acceptor for all donors P000180
c ndonr = number of identified donors P000190
c common/stkn/ nstak P000200
c common/stkp/ xstak(100),ystak(100),zstak P000210
c common/stkt/ vstak(100),tstak(100) P000220
c common/inst/ mstak(100) P000230
c initialize stack initiation statuses and times P000240
c do 110 istak=1,nstak P000250
c mstak(istak)=0 P000260
c tstak(istak)=1.e9 P000270
110 continue P000280
c initialize donor initiation status and time P000290
c ndonr=1 P000300
c mstak(idoni)=1 P000310
c tstak(idoni)=0. P000320
c find earliest acceptor for all donors P000330
120 tnext=1.e9 P000340
c cycle through the stacks P000350
c do 130 istak=1,nstak P000360
c if the current stack is an identified donor P000370
c if(mstak(istak).gt.0) then P000380
c determine the earliest acceptor for that donor P000390
c call arly(xstak(istak),ystak(istak),tstak(istak),vstak(istak), P000400
c + ierly,terly) P000410
c save the earliest acceptor for all donors P000420
c if(terly.lt.tnext) then P000430
c idonx=ierly P000440
c tnext=terly P000450
c endif P000460
c endif P000470
c 130 continue P000480
c increment donor P000490
c ndonr=ndonr+1 P000500
c assign earliest acceptor as next donor P000510
c mstak(idonx)=ndonr P000520
c tstak(idonx)=tnext P000530
c check for completion of stack initiation time assignments P000540
c if(ndonr.lt.nstak) go to 120 P000550
c return P000560
c end P000570
c----- P000580
c----- P000590
c----- P000600
c----- P000610
c----- P000620
c----- P000630
c----- P000640
c----- P000650
c----- P000660
c----- P000670
c----- P000680
c----- P000690
c----- P000700
c----- P000710
c----- P000720
c----- P000730
c----- P000740
c----- P000750
c----- P000760
c----- P000770
c----- P000780

```

Figure A-13. Subroutine prop from the propagation module.

```

c
c subroutine inpt
c obtains problem input from the file bwaco.inp
c problem variable:
c   ulabl = user's problem label
c
c stack variables:
c   nstak = total number of stacks
c   xstak = array of x-coordinates for the stacks
c   ystak = array of y-coordinates for the stacks
c   zstak = z-coordinate for the stacks
c   wstak = array of yields for the stacks
c   vstak = array of stack propagation velocities
c   idonr = array of initial donor switches
c
c region variables:
c   xmin = minimum x-coordinate of region of interest
c   xmax = maximum x-coordinate of region of interest
c   ymin = minimum y-coordinate of region of interest
c   ymax = maximum y-coordinate of region of interest
c
c common/ulab/ ulabl
c common/stkn/ nstak
c common/stkp/ xstak(100),ystak(100),zstak
c common/stkt/ vstak(100),wstak(100)
c common/stky/ wstak(100)
c common/stkd/ idonr(100),nidnr
c common/mxmn/ xmin,xmax,ymin,ymax
c
c character*80 ulabl,dummy
c
c   log call
c   write(8,3010)
c
c   open the user input file
c   open(7,file='bwaco.inp',form='formatted',status='old',err=140)
c
c   read the user's problem label
c   read(7,1010,end=160,err=150) ulabl
c   write(6,3020) ulabl
c
c   read the region data
c   read(7,1010,end=160,err=150) dummy
c   read(7,1020,end=160,err=150) xmin,xmax,ymin,ymax
c   write(6,3030) xmin,xmax,ymin,ymax
c
c   read the stack data
c   read(7,1010,end=160,err=150) dummy
c   nstak=0
c   zstak=0.
c   110 nstak=nstak+1
c     if(nstak.gt.100) go to 120
c     read(7,1020,end=120,err=150)
c   + xstak(nstak),ystak(nstak),wstak(nstak),vstak(nstak),idonr(nstak)
c   + write(8,3040)
c   , + xstak(nstak),ystak(nstak),wstak(nstak),vstak(nstak),idonr(nstak)
c   , go to 110
c
c   close the user input file and adjust the number of stacks
c   120 close(7)
c   nstak=nstak-1
c
c   remove any bogus stacks
c   130 continue
c     if(xstak(nstak).eq.0..and.ystak(nstak).eq.0..and.
c     + wstak(nstak).eq.0..and.vstak(nstak).eq.0..and.
c     + idonr(nstak).eq.0) then
c       nstak=nstak-1
c       go to 130
c     endif
c     if(nstak.lt.1) go to 170
c
c   write(8,3050) nstak
c
c   return
c
c   errors detected
c   140 stop ' BWACO: ERROR - Unable to open bwaco.inp!'
c   150 stop ' BWACO: ERROR - Unable to read bwaco.inp!'
c   160 stop ' BWACO: ERROR - Premature end of bwaco.inp!'
c   170 stop ' BWACO: ERROR - Fewer than one stack specified!'
c
c   format statements
c   1010 format(80)
c   1020 format(1f10.2,4x,i1)
c   3010 format(' inpt called')
c   3020 format(' label: ',a80)
c   3030 format(' region: ',4f10.2)
c   3040 format(' stack: ',4f10.2,i2)
c   3050 format(' stacks: ',i3)
c
c   end
c

```

Figure A-14. Subroutine inpt from the execution module.

```

c----- S000010
c----- S000020
c----- subroutine sseq(nstck,xstck,ystck,wstck, xsseq,ysseq,wsseq) S000030
c----- computes the single stack equivalent of the distributed stacks S000040
c----- independent variables: S000050
c----- nstck = number of stacks S000060
c----- xstck = array of x-coordinates for the stacks S000070
c----- ystck = array of y-coordinates for the stacks S000080
c----- wstck = array of yields for the stacks S000090
c----- dependent variables: S000100
c----- xsseq = x-coordinate of center of charge S000110
c----- ysseq = y-coordinate of center of charge S000120
c----- wsseq = single stack equivalent weight S000130
c----- local variable: S000140
c----- istck = stack counter S000150
c----- dimension xstck(100),ystck(100),wstck(100) S000160
c----- compute the single stack equivalent weight S000170
c----- wsseq=0. S000180
c----- do 110 istck=1,nstck S000190
c-----   wsseq=wsseq+wstck(istck) S000200
110 continue S000210
c----- compute the center of charge S000220
c----- xsseq=cent(nstck,xstck,wstck,wsseq) S000230
c----- ysseq=cent(nstck,ystck,wstck,wsseq) S000240
c----- return S000250
c----- end S000260
c----- S000270
c----- S000280
c----- S000290
c----- S000300
c----- S000310
c----- S000320
c----- S000330
c----- S000340
c----- S000350
c----- S000360
c----- C000010
c----- C000020
c----- C000030
c----- C000040
c----- C000050
c----- C000060
c----- C000070
c----- C000080
c----- C000090
c----- C000100
c----- C000110
c----- C000120
c----- C000130
c----- C000140
c----- C000150
c----- dimension rstck(100),wstck(100) C000160
c----- C000170
c----- rwstk=0. C000180
c----- do 110 istck=1,nstck C000190
c-----   rwstk=rwstk+rstck(istck)*wstck(istck) C000200
110 continue C000210
c----- cent=rwstk/wtotl C000220
c----- return C000230
c----- end C000240
c----- C000250
c----- C000260
c----- C000270

```

Figure A-15. Subroutine sseq and function cent from the execution module.

```

c
c subroutine stak
c defines the stacks
c
c stack variables:
c nstak = total number of stacks
c xstak = array of x-coordinates for the stacks
c ystak = array of y-coordinates for the stacks
c zstak = z-coordinate for the stacks
c wstak = array of yields for the stacks
c vstak = array of stack propagation velocities
c nidnr = number of initial donors specified
c
c region variables:
c xcent = x-coordinate of center of charge
c ycent = y-coordinate of center of charge
c rinhb = inhabited building radius
c xmini = minimum x-coordinate
c xmaxi = maximum x-coordinate
c ymini = minimum y-coordinate
c ymaxi = maximum y-coordinate
c
c local variables:
c istak = stack counter
c dregn = region margin
c rregn = region radius
c
c common/conf/ thrd, tpi
c common/conv/ okt, ocm, odsc, ocmtk3
c common/regn/ xcent, ycent, rinhb, xmini, xmaxi, ymini, ymaxi
c common/alow/ pallo
c common/stkn/ hstak
c common/stkp/ xstak(100), ystak(100), zstak
c common/stky/ wstak(100)
c common/stkt/ vstak(100), tstak(100)
c common/stkd/ idonr(100), nidnr
c common/totl/ wtotl
c
c log call
c write(8,3010)
c
c compute single charge equivalent and inhabited building radius
c call ssed(nstak,xstak,ystak,wstak, xcent,ycent,wtotl)
c if(wtotl.lt.3.0e4) then
c   if(wtotl.ge.100.) then
c     rinhb=380.
c   else
c     rinhb=204.
c   endif
c   pallo=8.3e4
c else
c   if(wtotl.ge.3.0e4.and.wtotl.lt.1.0e5) then
c     rinhb=1.2*wtotl**thrd
c     pallo=8.3e4
c   endif
c   if(wtotl.ge.1.0e5.and.wtotl.lt.2.5e5) then
c     rinhb=0.738*wtotl**0.577
c     pallo=9.7e4-0.14*wtotl
c   endif
c   if(wtotl.ge.2.5e5) then
c     rinhb=15.24*wtotl**thrd
c     pallo=6.2e4
c   endif
c   dregn=0.05*rinhb
c   rregn=rinhb+dregn
c   xmini=xcent-rregn
c   xmaxi=xcent+rregn
c   ymini=ycent-rregn
c   ymaxi=ycent+rregn
c   write(8,3020) rinhb,xmini,xmaxi,ymini,ymaxi
c
c adjust the region limits, convert the stack variables to 1-kt
c standard units and double the yields for ground plane simulation
c nidnr=0
c do 110 istak=1,nstak
c   rstak=15.24*wstak(istak)**thrd+dregn
c   xmini=amin1(xstak(istak)-rstak,xmini)
c   xmaxi=amax1(xstak(istak)+rstak,xmaxi)
c   ymini=amin1(ystak(istak)-rstak,ymini)
c   ymaxi=amax1(ystak(istak)+rstak,ymaxi)
c   xstak(istak)=xstak(istak)/ocm
c   ystak(istak)=ystak(istak)/ocm
c   wstak(istak)=2.*wstak(istak)/okt
c   vstak(istak)=vstak(istak)/ocm
c   write(8,3030)
c   + xstak(istak), ystak(istak), wstak(istak), vstak(istak)
c   if(idonr(istak).ne.0) nidnr=nidnr+1
c 110 continue
c
c check the number of initial donors
c if(nidnr.eq.0)
c   + stop 'BWACO: ERROR - No initial donors were specified!'
c if(nidnr.lt.nstak) write(6,2010)
c write(8,3040) nidnr
c
c return
c
c format statements
c 2010 format('' BWACO: WARNING - Some stacks are not initial donors!'')
c 3010 format('' stak called'')
c 3020 format('' region: ',1b5e10.3)
c 3030 format('' stack: ',4f10.2)
c 3040 format('' donors: ',13)
c
c end
c

```

Figure A-16. Subroutine stak from the execution module.

```

c
c----- subroutine blok(lbmax,nstat,argiv, lblok,nblk)
c
c determines the integer block length and number
c of blocks that yields the block aspect ratio
c that most closely matches a given aspect ratio
c
c independent variables:
c   lbmax = maximum block length
c   nstat = number of stations
c   argiv = given aspect ratio
c
c dependent variables:
c   lblok = length of blocks
c   nblk = number of blocks
c
c determine the minimum block length
c   lbmin=nstat/lbmax
c
c initialize the counter
c   numbr=0
c
c cycle through all possible block lengths
c do 110 lblok=lbmin,lbmax
c
c   if there are an integral number of blocks of length lblok
c   if(mod(nstat,lblok).eq.0) then
c
c     determine the number of blocks
c     nblk=nstat/lblok
c
c     determine the block aspect ratio
c     arblik=float(lblk)/float(nblk)
c
c     as soon as the block aspect ratio
c     exceeds the given aspect ratio
c     if(arblk.ge.argiv) then
c
c       if a previous solution has been determined
c       if(numbr.gt.0) then
c
c         compute the differences between the given aspect
c         ratio and the previous and current aspect ratios
c         dold=abs(argiv-arblo)
c         dcur=abs(argiv-arblk)
c
c         if the given aspect ratio is
c         closer to the previous aspect ratio
c         if(dold.lt.dcur) then
c
c           return the block length corresponding
c           to the previous block aspect ratio
c           lblok=lblok0
c           nblk=nblk0
c
c         otherwise
c         else
c
c           return the block length corresponding
c           to the current block aspect ratio
c           lblok=lblok
c           nblk=nblk
c
c         endif
c
c       otherwise
c       else
c
c         return the block length corresponding
c         to the current block aspect ratio
c         lblok=lblok
c         nblk=nblk
c
c       endif
c
c       return
c
c     endif
c
c     increment the counter and save the current values
c     numbr=numbr+1
c     lblok0=lblok
c     nblk0=nblk
c     arblo=arblk
c
c   endif
c
c 110 continue
c
c   if a previous solution exists
c   if(numbr.gt.0) then
c
c     return the block length corresponding
c     to the previous block aspect ratio
c     lblok=lblok0
c     nblk=nblk0
c     return
c
c   otherwise
c   else
c
c     an error condition exists
c     stop 'BWACO: ERROR - Region cannot be blocked!'
c
c   endif
c
c end
c

```

Figure A-17. Subroutine blok from the execution module.

```

c
c subroutine stns
c defines the stations
c
c station variables:
c   nstat = number of stations
c   lblock = length of station blocks
c   nblok = number of station blocks
c   xstat = array of x-coordinates for the stations
c   ystat = array of y-coordinates for the stations
c   zstat = z-coordinate of stations
c   icoal = array of coalescence statuses for the stations
c   wstat = array of effective yields for the stations
c   pstat = array of effective peak overpressures for the stations
c
c region variables:
c   rinhb = inhabited building radius
c   xmin = minimum x-coordinate
c   xmax = maximum x-coordinate
c   ymin = minimum y-coordinate
c   ymax = maximum y-coordinate
c
c local variables:
c   istan = station index
c   istan = station index
c   istat = station counter
c   xregn = x-dimension of region
c   yregn = y-dimension of region
c   asrat = aspect ratio of region
c   deltx = x-increment
c   delty = y-increment
c
c common/conv/ okt,ocm,odsc,ocmkt3
c common/icfg/ lcomb,iscrn,iprnt
c common/mxmb/ xmin,xmax,ymin,ymax
c common/regn/ xcent,vcent,rinhb,xmin,xmaxi,ymini,ymaxi
c common/ambi/ pambi,rambi
c common/sttb/ hstat
c common/sttb/ lblock,nblock
c common/sttb/ xstat(120),ystat(120),zstat
c common/sttb/ icoal(3600),pstat(3600),wstat(3600)
c
c log call
c write(8,3010)
c
c check the region limits
c if(xmin.eq.xmax) then
c   xmin=xmin
c   xmax=xmax
c endif
c if(ymin.eq.ymax) then
c   ymin=ymin
c   ymax=ymax
c endif
c write(8,3020) xmin,xmax,ymin,ymax
c
c convert the region limits to 1-kt standard units
c xmin=xmin/ocm
c xmax=xmax/ocm
c ymin=ymin/ocm
c ymax=ymax/ocm
c write(8,3020) xmin,xmax,ymin,ymax
c
c define the number of stations
c nstat=3600
c lblock=120
c
c compute the coordinate ranges for the region
c xregn=xmax-xmin
c yregn=ymax-ymin
c
c distribute the station indices
c asrat=xregn/yregn
c call blo(1block,nstat,asrat, lblock,nblock)
c write(8,3030) lblock,nblock
c
c define the x values
c deltx=xregn/float(lblock)
c do 110 istan=1,lblock
c   xstat(istan)=(float(istan-1)+0.5)*deltx+xmin
c 110 continue
c
c define the y values
c delty=yregn/float(nblock)
c do 120 jstan=1,nblock
c   ystat(jstan)=(float(jstan-1)+0.5)*delty+ymin
c 120 continue
c
c zstat=0.
c
c initialize the coalescence status, effective yield,
c overpressure and ambient pressure and density
c do 130 istat=1,nstat
c   icoal(istat)=0
c   wstat(istat)=0.
c   pstat(istat)=0.
c 130 continue
c call at62(0., pambi,dummy,rambi,dummy)
c
c return
c
c format statement
c 3010 format(' stns called')
c 3020 format(' region:',4f10.2)
c 3030 format(' blocks:',2i4)
c
c end
c

```

Figure A-18. Subroutine stns from the execution module.

```

c_____ R000010
c_____ R000020
c subroutine rtim(ihbgn,imbgn,isbgn,ilbgn,ihend,imend,isend,ilend) R000030
c_____ R000040
c determines and reports computation time R000050
c_____ R000060
c independent variables: R000070
c_____ ihbgn = hour at which computation begins R000080
c_____ imbgn = minute at which computation begins R000090
c_____ isbgn = second at which computation begins R000100
c_____ ilbgn = tenth second at which computation begins R000110
c_____ ihend = hour at which computation ends R000120
c_____ imend = minute at which computation ends R000130
c_____ isend = second at which computation ends R000140
c_____ ilend = tenth second at which computation ends R000150
c_____ R000160
c local variables: R000170
c_____ tsbgn = time at which computation begins R000180
c_____ tsend = time at which computation ends R000190
c_____ tscmp = computation time R000200
c_____ R000210
c compute the computation time R000220
c_____ if(ihend.lt.ihbgn) ihend=ihend+24 R000230
c_____ tsbgn=3600.*float(ihbgn)+60.*float(imbgn) R000240
c_____ +(float(isbgn)+float(ilbgn)/100.) R000250
c_____ tsend=3600.*float(ihend)+60.*float(imend) R000260
c_____ +(float(isend)+float(ilend)/100.) R000270
c_____ tscmp=tsend-tsbgn R000280
c_____ R000290
c report the computation time R000300
c_____ if(tscmp.lt.3600.) then R000310
c_____ if(tscmp.lt.60.) then R000320
c_____ write(6,2010) tscmp R000330
c_____ else R000340
c_____ write(6,2020) tscmp/60. R000350
c_____ endif R000360
c_____ else R000370
c_____ write(6,2030) tscmp/3600. R000380
c_____ endif R000390
c_____ return R000400
c format statements R000420
c_____ 2010 format('' Computation Time =',f7.2,' seconds'') R000440
c_____ 2020 format('' Computation Time =',f7.2,' minutes'') R000450
c_____ 2030 format('' Computation Time =',f7.2,' hours'') R000460
c_____ R000470
c end R000480
c_____ R000490

```

Figure A-19. Subroutine *rtim* from the execution module.

Subroutine *copt* is shown in Figure A-20. It cycles through all possible donors (as specified by user input) computing the initiation times with calls to *prop* and the effective yields with calls to *effy*. It also obtains the beginning and ending times of the computations and calls *rtim* to compute and report the run time.

PLOTTING MODULE

These routines plot the results of the computations on the request of the user using proprietary graphics routines.

Subroutine *rest*, shown in Figure A-21, restores the original units and eliminates the ground plane factor.

Subroutine *spmn*, shown in Figure A-22, displays the BWACO plotting menu and obtains the user's request.

Subroutine *plts*, which is not listed, generates the coalescence and peak overpressure maps using calls to proprietary routines.

INTERFACE MODULE

These routines set up the session and direct the main BWACO functions at the discretion of the user.

Subroutine *smmn*, shown in Figure A-23, displays the BWACO main menu and obtains the user's request.

Subroutine *exec*, shown in Figure A-27, calls the routines of the execution module when the user requests that a problem be run.

Subroutine *rcfg*, shown in Figure A-24, reconfigures the session, allowing the user to select the default display and printer.

```

c-----C000010
c-----C000020
c-----C000030
c-----C000040
c-----C000050
c-----C000060
c-----C000070
c-----C000080
c-----C000090
c-----C000100
c-----C000110
c-----C000120
c-----C000130
c-----C000140
c-----C000150
c-----C000160
c-----C000170
c-----C000180
c-----C000190
c-----C000200
c-----C000210
c-----C000220
c-----C000230
c-----C000240
c-----C000250
c-----C000260
c-----C000270
c-----C000280
c-----C000290
c-----C000300
c-----C000310
c-----C000320
c-----C000330
c-----C000340
c-----C000350
c-----C000360
c-----C000370
c-----C000380
c-----C000390
c-----C000400
c-----C000410
c-----C000420
c-----C000430
c-----C000440
c-----C000450
c-----C000460
c-----C000470
c-----C000480
c-----C000490
c-----C000500
c-----C000510
c-----C000520
c-----C000530
c-----C000540
c-----C000550
c-----C000560
c-----C000570
c-----C000580
c-----C000590
c-----C000600
c-----C000610
c-----C000620
c-----C000630
c-----C000640
c-----C000650
c
c subroutine copt
c
c determines the worst-case coalescence pattern
c
c stack variables:
c   nstak = total number of stacks
c   idonr = array of initial donor switches
c
c local variables:
c   istak = stack counter
c   ihbgn = hour at which computation begins
c   imbgn = minute at which computation begins
c   isbgn = second at which computation begins
c   ilbgn = tenth second at which computation begins
c   ihend = hour at which computation ends
c   imend = minute at which computation ends
c   isend = second at which computation ends
c   ilend = tenth second at which computation ends
c
c common/stkn/ nstak
c common/stkd/ idonr(100),nidnr
c
c   log call
c   write(8,3010)
c
c   obtain starting time
c   call gettim(ihbgn,imbgn,isbgn,ilbgn)
c
c   cycle through the stacks
c   idcnt=0
c   do 110 istak=1,nstak
c
c     if the stack is a possible initial donor
c     if(idonr(istak).eq.1) then
c
c       report the current initial donor number
c       idcnt=idcnt+1
c       write(6,2010) idcnt,nidnr
c
c       determine the propagation sequence
c       call prop(istak)
c
c       determine the (worst-case) effective yields
c       call effy
c
c     endif
c
c   110 continue
c
c   obtain completion time
c   call gettim(ihend,imend,isend,ilend)
c
c   determine and report computation time
c   call rtim(ihbgn,imbgn,isbgn,ilbgn,ihend,imend,isend,ilend)
c
c   return
c
c   format statements
2010 format( /' Initial Donor ',i3,' of ',i3,':')
3010 format(/' copt called')
c
c   end
c

```

Figure A-20. Subroutine copt from the execution module.

```

c-----R000010
c-----R000020
c-----R000030
c-----R000040
c-----R000050
c-----R000060
c-----R000070
c-----R000080
c-----R000090
c-----R000100
c-----R000110
c-----R000120
c-----R000130
c-----R000140
c-----R000150
c-----R000160
c-----R000170
c-----R000180
c-----R000190
c-----R000200
c-----R000210
c-----R000220
c-----R000230
c-----R000240
c-----R000250
c-----R000260
c-----R000270
c-----R000280
c-----R000290
c-----R000300
c-----R000310
c-----R000320
c-----R000330
c-----R000340
c-----R000350
c-----R000360
c-----R000370
c-----R000380
c-----R000390
c-----R000400
c-----R000410
c-----R000420
c-----R000430
c-----R000440
c-----R000450
c-----R000460
c-----R000470
c-----R000480
c-----R000490
c-----R000500
c-----R000510
c-----R000520
c-----R000530
c-----R000540
c-----R000550
c-----R000560
c-----R000570
c-----R000580
c-----R000590
c-----R000600
c-----R000610
c
c   subroutine rest
c
c   restores original units and plots the results
c
c   stack variables:
c       nstak = number of stacks
c       xstak = array of x-coordinates for the stacks
c       ystak = array of y-coordinates for the stacks
c
c   station variables:
c       lblok = length of station blocks
c       nblok = number of station blocks
c       nstat = number of stations
c       wstat = array of effective yields for the stations
c
c   local variables:
c       istak = stack counter
c       istan = station index
c       jstan = station index
c       istat = station counter
c
c   common/conv/  okt,ocm,odsc,ocmkt3
c   common/stkn/  nstak
c   common/stkp/  xstak(100),ystak(100),zstak
c   common/sttb/  lblok,nblok
c   common/sttn/  nstat
c   common/sttp/  xstat(120),ystat(120),zstat
c   common/sttv/  icoal(3600),pstat(3600),wstat(3600)
c
c   log call
c   write(8,3010)
c
c   restore the original units to the stack coordinates
c   do 110 istak=1,nstak
c       xstak(istak)=ocm*xstak(istak)
c       ystak(istak)=ocm*ystak(istak)
c 110 continue
c
c   restore the original units to the station coordinates
c   do 120 istan=1,lblok
c       xstat(istan)=ocm*xstat(istan)
c 120 continue
c   do 130 jstan=1,nblok
c       ystat(jstan)=ocm*ystat(jstan)
c 130 continue
c
c   restore the original units to and eliminate the
c   ground plane factor from the effective yields
c   do 140 istat=1,nstat
c       wstat(istat)=0.5*okt*wstat(istat)
c 140 continue
c
c   return
c
c   format statement
c   3010 format(' rest called')
c
c   end
c

```

Figure A-21. Subroutine rest from the plotting module.

```

c-----S000010
c-----S000020
c-----S000030
c-----S000040
c-----S000050
c-----S000060
c-----S000070
c-----S000080
c-----S000090
c-----S000100
c-----S000110
c-----S000120
c-----S000130
c-----S000140
c-----S000150
c-----S000160
c-----S000170
c-----S000180
c-----S000190
c-----S000200
c-----S000210
c-----S000220
c-----S000230
c-----S000240
c-----S000250
c-----S000260
c-----S000270
c-----S000280
c-----S000290
c-----S000300
c-----S000310
c-----S000320
c-----S000330
c-----S000340
c-----S000350
c-----S000360
c-----S000370
c-----S000380
c-----S000390
c-----S000400
c-----S000410
c-----S000420
c-----S000430
c-----S000440
c-----S000450
c-----S000460
c-----S000470
c-----S000480
c-----S000490
c-----S000500
c-----S000510
c-----S000520
c-----S000530
c-----S000540
c-----S000550
c-----S000560
c-----S000570
c-----S000580
c-----S000590
c-----S000600
c
c-----subroutine spmn(irqst, iptyp, idevi)
c-----displays plotting menu and obtains user request
c-----dependent variables:
c-----    irqst = user request index
c-----    iptyp = index of requested plot type
c-----    idevi = index of requested device
c-----log call
c-----    write(8,3010)
c-----display menu
c-----    write(6,2010)
110  write(6,2020)
c-----    write(6,2030)
c-----    write(6,2040)
c-----    write(6,2050)
c-----    write(6,2060)
c-----    write(6,2070)
c-----    write(6,2080)
c-----obtain user request
c-----    read(5,1010,err=110) irqst
c-----    if(irqst.lt.0.orirqst.gt.5) go to 110
c-----    write(8,3020) irqst
c-----assign requested plot type
c-----    if(irqst.eq.1.or.irqst.eq.2) then
c-----        iptyp=2
c-----    else
c-----        iptyp=1
c-----    endif
c-----assign requested device
c-----    if(irqst.eq.1.or.irqst.eq.3) then
c-----        idevi=iscrn
c-----    else
c-----        idevi=iprnt
c-----    endif
c-----return
c-----format statements
1010 format(i1)
2010 format(//, 'BWACO PLOTTING MENU')
2020 format( /, ' Enter 1 to plot Coalescence Map (Screen)')
2030 format( /, ' 2 to plot Coalescence Map (Printer)')
2040 format( /, ' 3 to plot Peak Overpressure Map (Screen)')
2050 format( /, ' 4 to plot Peak Overpressure Map (Printer)')
2060 format( /, ' 5 to reconfigure the session')
2070 format( /, ' 0 to return to the Main Menu')
2080 format( /, ' \')
3010 format(//, ' spmn called')
3020 format(' ', 'irqst='i1)
c-----end
c

```

Figure A-22. Subroutine spmn from the plotting module.

```

c----- S000010
c----- S000020
c----- subroutine smmn(iscur,irqst) S000030
c----- displays main menu and obtains user request S000040
c----- display menu heading S000050
c----- write(6,2010) S000060
c----- check for presence of bwaco.dat S000070
c----- open(7,file='bwaco.dat',status='old',err=120) S000080
c----- close(7) S000090
c----- display complete menu S000100
110  write(6,2020) S000110
      if(iscur.eq.0) then S000120
          write(6,2030) S000130
      else S000140
          write(6,2031) S000150
      endif S000160
      write(6,2041) S000170
      write(6,2050) S000180
      write(6,2060) S000190
c----- obtain user request S000200
c----- read(5,1010,err=110) irqst S000210
c----- check for valid request S000220
c----- if(irqst.lt.0.orirqst.gt.3) go to 110 S000230
c----- go to 130 S000240
c----- display partial menu S000250
120  write(6,2020) S000260
      write(6,2042) S000270
      write(6,2050) S000280
      write(6,2060) S000290
c----- obtain user request S000300
c----- read(5,1010,err=120) irqst S000310
c----- check for valid request S000320
c----- if(irqst.lt.0.orirqst.gt.2) go to 120 S000330
c----- adjust request number S000340
c----- if(irqst.eq.2) irqst=3 S000350
c----- 130  return S000360
c----- format statements S000370
1010 format(i1) S000380
2010 format('' BWACO MAIN MENU '') S000390
2020 format('' Enter 1 to run the current problem') S000400
2030 format('' 2 to plot the previous problem') S000410
2031 format('' 2 to plot the current problem') S000420
2041 format('' 3 to reconfigure the session') S000430
2042 format('' 2 to reconfigure the session') S000440
2050 format('' 0 to end the session') S000450
2060 format('' '') S000460
c----- end S000470
c----- S000480
c----- S000490
c----- S000500
c----- S000510
c----- S000520
c----- S000530
c----- S000540
c----- S000550
c----- S000560
c----- S000570
c----- S000580
c----- S000590
c----- S000600
c----- S000610
c----- S000620

```

Figure A-23. Subroutine smmn from the interface module.

```

c
c subroutine rcfg(icfm,ifrom)
c
c reconfigures the session and rewrites the bwaco.cfg file
c
c independent variables:
c   icfm = mandatory configuration save switch
c   ifrom = configuration file index
c
c configuration variables:
c   iocm = pressure combination algorithm switch
c   iscrn = default printer switch
c   iprint = default print switch
c
c common/icfg/ iocm,iscrn,iprint
c
c character*14 lable,labip
c
c
c   l10 call
c   write(1,1010)
c
c   110 assign the current default display to the menu heading
c
c   l110 lable = 'CGA Mode 4 320x200/16'
c   l111 lable = 'CGA Mode 4 320x200/16'
c   l112 lable = 'CGA Mode 4 320x200/16'
c   l113 lable = 'CGA Mode 4 320x200/16'
c   l114 lable = 'CGA Mode 4 320x200/16'
c   l115 lable = 'CGA Mode 4 320x200/16'
c   l116 lable = 'CGA Mode 4 320x200/16'
c   l117 lable = 'CGA Mode 4 320x200/16'
c   l118 lable = 'CGA Mode 4 320x200/16'
c   l119 lable = 'CGA Mode 4 320x200/16'
c   l120 lable = 'CGA Mode 4 320x200/16'
c
c   120 assign the current default printer to the menu heading
c
c   l121 lable = 'Single-density printer'
c   l122 lable = 'Double-density printer'
c   l123 lable = 'Epson Print' 'Plot mode'
c   l124 lable = 'HP Plotter (Scaling)'
c   l125 lable = 'HP Plotter (Absolute)'
c   l126 lable = 'HP Laser Jet (75 dpi)'
c   l127 lable = 'HP Laser Jet (100 dpi)'
c   l128 lable = 'HP Laser Jet (150 dpi)'
c   l129 lable = 'HP Laser Jet (200 dpi)'
c   l130 lable = 'HP Think Jet Hi Dens'
c   l131 lable = 'HP Paint Jet Hi Dens'
c   l132 lable = 'HP Paint Jet Lo Dens'
c   l133 lable = 'Encapsulated Postscript'
c
c   130 display the current selections
c   write(6,2030) lable
c   write(6,2030) labip
c
c   120 display the configuration menu
c   write(6,2030)
c   if(ifcm.eq.5) write(6,2060)
c   if(ifcm.eq.3) write(6,2071)
c   if(ifcm.eq.2) write(6,2072)
c   write(6,2073)
c
c   130 obtain the user request
c   read(5,1010,err=120) lchng
c
c   check the validity of the request
c   if(lchng.lt.0.or.lchng.gt.120) go to 120
c
c   if(lchng.eq.1) then
c
c   process a request to change the default display
c   write(6,2130)
c   write(6,2140)
c   write(6,2140)
c   write(6,2140)
c   write(6,2140)
c
c   130 obtain the requested display
c   read(5,1020,err=130) lscrn
c
c   check the validity of the request
c   if(lscrn.lt.1.or.lscrn.gt.13) go to 130
c
c   130 set the requested display
c   write(8,3020) lscrn
c
c   endif
c
c   if(lchng.eq.2) then
c
c   process a request to change the default printer
c   write(6,2150)
c   write(6,2160)
c   write(6,2160)
c   write(6,2160)
c   write(6,2160)
c
c   140 obtain the requested printer
c   read(5,1020,err=140) iprint
c
c   check the validity of the request
c   if(iprint.lt.1.or.iprint.gt.140) go to 140
c
c   140 set the requested printer
c   write(8,3030) iprint
c
c   endif
c
c   if(ifcm.eq.1.or.lchng.eq.3) then
c
c   process a request to save the configuration
c   write(7,1110) 'Save configuration',err=170
c   close(7)
c
c   150 save
c   write(6,3040)
c
c   endif
c
c   if(lchng.ne.0) go to 110
c
c   return
c
c   error detected
c   170 stop 'BWACO: ERROR - Unable to open bwaco.cfg'
c
c   format statements
c
c   1010 format(1x)
c   1020 format(1x,212)
c   1030 format(1x,'BWACO CONFIGURATION MENU')
c   1040 format(1x,'Default Display: ',122)
c   1050 format(1x,'Enter 1 to change the default display')
c   1060 format(1x,'Enter 2 to change the default printer')
c   1070 format(1x,'Enter 3 to save the configuration')
c   1080 format(1x,'Enter 4 to return to the Main Menu')
c   1090 format(1x,'Enter 5 to return to the floating menu')
c   1100 format(1x,'CGA Mode 4 320x200/4 2: CGA Mode 3 320x200/16')
c   1120 format(1x,'CGA Mode 4 640x400/16 4: Hercules 720x340/16')
c   1130 format(1x,'5: EGA Mode 13 320x200/16 6: EGA Mode 14 640x200/16')
c   1140 format(1x,'7: EGA Mode 15 640x350/16 8: EGA Mode 16 640x350/16')
c   1150 format(1x,'9: VGA Mode 17 640x400/16 10: VGA Mode 18 640x400/16')
c   1160 format(1x,'11: ATI Mode 31 640x400/16 12: ATI Mode 32 720x400/16')
c   1170 format(1x,'13: EGA Mode 18 640x400/16 14: EGA Mode 19 720x400/16')
c   1180 format(1x,'15: Single-density printer 16: Double-density printer')
c   1220 format(1x,'Double-density printer 17: Epson Print plot mode')
c   1230 format(1x,'18: HP Plotter (Scaling) 19: HP Plotter (Absolute)')
c   1240 format(1x,'20: HP Laser Jet (75 dpi) 21: HP Laser Jet (100 dpi)')
c   1250 format(1x,'22: HP Laser Jet (150 dpi) 23: HP Laser Jet (200 dpi)')
c   1260 format(1x,'24: HP Think Jet Hi Dens 25: HP Think Jet Hi Dens')
c   1270 format(1x,'26: HP Paint Jet Lo Dens 27: HP Paint Jet Hi Dens')
c   1280 format(1x,'28: HP Paint Jet Hi Dens')
c   1290 format(1x,'29: Encapsulated Postscript')
c
c   1370 format(1x,'Enter the new printer index: ')
c   1380 format(1x,' 1010 1020 1030 1040 1050 1060 1070 1080 1090 1095 1100 1110 1120 1130 1140 1150 1160 1170 1180 1190 1200 1210 1220 1230 1240 1250 1260 1270 1280 1290 1300 1310 1320 1330 1340 1350 1360 1370 1380 1390 1400 1410 1420 1430 1440 1450 1460 1470 1480 1490 1500 1510 1520 1530 1540 1550 1560 1570 1580 1590 1600 1610 1620 1630 1640 1650 1660 1670 1680 1690 1700 1710 1720 1730 1740 1750 1760 1770 1780 1790 1800 1810 1820 1830 1840 1850 1860 1870 1880 1890 1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 1995 2000 2010 2020 2030 2040 2050 2060 2070 2080 2090 2095 2100 2110 2120 2130 2140 2150 2160 2170 2180 2190 2195 2200 2210 2220 2230 2240 2250 2260 2270 2280 2290 2300 2310 2320 2330 2340 2350 2360 2370 2380 2390 2395 2400 2410 2420 2430 2440 2450 2460 2470 2480 2490 2495 2500 2510 2520 2530 2540 2550 2560 2570 2580 2590 2595 2600 2610 2620 2630 2640 2650 2660 2670 2680 2690 2695 2700 2710 2720 2730 2740 2750 2760 2770 2780 2790 2795 2800 2810 2820 2830 2840 2850 2860 2870 2880 2890 2895 2900 2910 2920 2930 2940 2950 2960 2970 2980 2990 2995 3000 3010 3020 3030 3040 3050 3060 3070 3080 3090 3095 3100 3110 3120 3130 3140 3150 3160 3170 3180 3190 3195 3200 3210 3220 3230 3240 3250 3260 3270 3280 3290 3295 3300 3310 3320 3330 3340 3350 3360 3370 3380 3390 3395 3400 3410 3420 3430 3440 3450 3460 3470 3480 3490 3495 3500 3510 3520 3530 3540 3550 3560 3570 3580 3590 3595 3600 3610 3620 3630 3640 3650 3660 3670 3680 3690 3695 3700 3710 3720 3730 3740 3750 3760 3770 3780 3790 3795 3800 3810 3820 3830 3840 3850 3860 3870 3880 3890 3895 3900 3910 3920 3930 3940 3950 3960 3970 3980 3990 3995 4000 4010 4020 4030 4040 4050 4060 4070 4080 4090 4095 4100 4110 4120 4130 4140 4150 4160 4170 4180 4190 4195 4200 4210 4220 4230 4240 4250 4260 4270 4280 4290 4295 4300 4310 4320 4330 4340 4350 4360 4370 4380 4390 4395 4400 4410 4420 4430 4440 4450 4460 4470 4480 4490 4495 4500 4510 4520 4530 4540 4550 4560 4570 4580 4590 4595 4600 4610 4620 4630 4640 4650 4660 4670 4680 4690 4695 4700 4710 4720 4730 4740 4750 4760 4770 4780 4790 4795 4800 4810 4820 4830 4840 4850 4860 4870 4880 4890 4895 4900 4910 4920 4930 4940 4950 4960 4970 4980 4990 4995 5000 5010 5020 5030 5040 5050 5060 5070 5080 5090 5095 5100 5110 5120 5130 5140 5150 5160 5170 5180 5190 5195 5200 5210 5220 5230 5240 5250 5260 5270 5280 5290 5295 5300 5310 5320 5330 5340 5350 5360 5370 5380 5390 5395 5400 5410 5420 5430 5440 5450 5460 5470 5480 5490 5495 5500 5510 5520 5530 5540 5550 5560 5570 5580 5590 5595 5600 5610 5620 5630 5640 5650 5660 5670 5680 5690 5695 5700 5710 5720 5730 5740 5750 5760 5770 5780 5790 5795 5800 5810 5820 5830 5840 5850 5860 5870 5880 5890 5895 5900 5910 5920 5930 5940 5950 5960 5970 5980 5990 5995 6000 6010 6020 6030 6040 6050 6060 6070 6080 6090 6095 6100 6110 6120 6130 6140 6150 6160 6170 6180 6190 6195 6200 6210 6220 6230 6240 6250 6260 6270 6280 6290 6295 6300 6310 6320 6330 6340 6350 6360 6370 6380 6390 6395 6400 6410 6420 6430 6440 6450 6460 6470 6480 6490 6495 6500 6510 6520 6530 6540 6550 6560 6570 6580 6590 6595 6600 6610 6620 6630 6640 6650 6660 6670 6680 6690 6695 6700 6710 6720 6730 6740 6750 6760 6770 6780 6790 6795 6800 6810 6820 6830 6840 6850 6860 6870 6880 6890 6895 6900 6910 6920 6930 6940 6950 6960 6970 6980 6990 6995 7000 7010 7020 7030 7040 7050 7060 7070 7080 7090 7095 7100 7110 7120 7130 7140 7150 7160 7170 7180 7190 7195 7200 7210 7220 7230 7240 7250 7260 7270 7280 7290 7295 7300 7310 7320 7330 7340 7350 7360 7370 7380 7390 7395 7400 7410 7420 7430 7440 7450 7460 7470 7480 7490 7495 7500 7510 7520 7530 7540 7550 7560 7570 7580 7590 7595 7600 7610 7620 7630 7640 7650 7660 7670 7680 7690 7695 7700 7710 7720 7730 7740 7750 7760 7770 7780 7790 7795 7800 7810 7820 7830 7840 7850 7860 7870 7880 7890 7895 7900 7910 7920 7930 7940 7950 7960 7970 7980 7990 7995 8000 8010 8020 8030 8040 8050 8060 8070 8080 8090 8095 8100 8110 8120 8130 8140 8150 8160 8170 8180 8190 8195 8200 8210 8220 8230 8240 8250 8260 8270 8280 8290 8295 8300 8310 8320 8330 8340 8350 8360 8370 8380 8390 8395 8400 8410 8420 8430 8440 8450 8460 8470 8480 8490 8495 8500 8510 8520 8530 8540 8550 8560 8570 8580 8590 8595 8600 8610 8620 8630 8640 8650 8660 8670 8680 8690 8695 8700 8710 8720 8730 8740 8750 8760 8770 8780 8790 8795 8800 8810 8820 8830 8840 8850 8860 8870 8880 8890 8895 8900 8910 8920 8930 8940 8950 8960 8970 8980 8990 8995 9000 9010 9020 9030 9040 9050 9060 9070 9080 9090 9095 9100 9110 9120 9130 9140 9150 9160 9170 9180 9190 9195 9200 9210 9220 9230 9240 9250 9260 9270 9280 9290 9295 9300 9310 9320 9330 9340 9350 9360 9370 9380 9390 9395 9400 9410 9420 9430 9440 9450 9460 9470 9480 9490 9495 9500 9510 9520 9530 9540 9550 9560 9570 9580 9590 9595 9600 9610 9620 9630 9640 9650 9660 9670 9680 9690 9695 9700 9710 9720 9730 9740 9750 9760 9770 9780 9790 9795 9800 9810 9820 9830 9840 9850 9860 9870 9880 9890 9895 9900 9910 9920 9930 9940 9950 9960 9970 9980 9985 9990 9995 9999

```

Figure A-24. Subroutine *rcfg* from the interface module.

```

c-----          S000010
c subroutine savn          S000020
c   saves new data to file          S000030
c
c   common/jcfg/  icomb,iscrn,iprnt          S000040
c   common/totl/  wtotl          S000050
c   common/ulab/  ulabl          S000060
c   common/stkn/  nstat          S000070
c   common/stkp/  xstak(100),ystak(100),zstak          S000080
c   common/stkd/  idonr(100),nidnr          S000090
c   common/regn/  xcent,ycent,rinhb,xmini,xmaxi,ymini,ymaxi          S000100
c   common/stfn/  nstat          S000110
c   common/sttb/  lblok,nblk          S000120
c   common/sttp/  xstat(120),ystat(120),zstat          S000130
c   common/sttv/  icoal(3600),pstat(3600),wstat(3600)          S000140
c   common/mxmn/  xmin,xmax,ymin,ymax          S000150
c
c   character*80 ulabl          S000160
c
c   log call          S000170
c   write(8,3010)          S000180
c
c   open(7,file='bwaco.dat',form='unformatted',err=110)          S000190
c   write(7,ulabl)          S000200
c   write(7,nstat,lblok,nblk,nstat,nidnr)          S000210
c   write(7,wtotl,xmin,xmax,ymin,ymax)          S000220
c   write(7,xcent,ycent,rinhb)          S000230
c   write(7,(xstak(i),i=1,nstat))          S000240
c   write(7,(ystak(i),i=1,nstat))          S000250
c   write(7,(idonr(i),i=1,nstat))          S000260
c   write(7,(xstat(i),i=1,lblok))          S000270
c   write(7,(ystat(i),i=1,nblk))          S000280
c   write(7,(pstat(i),i=1,nstat))          S000290
c   write(7,(wstat(i),i=1,nstat))          S000300
c   write(7,(icoal(i),i=1,nstat))          S000310
c   close(7)          S000320
c
c   return          S000330
c
c   error detected          S000340
c   110 stop, BWACO: ERROR - Unable to open bwaco.dat!          S000350
c
c   format statement          S000360
c   3010 format(/' savn called')          S000370
c
c   end          S000380
c
c-----          S000390
c subroutine geto          S000400
c   gets old plotting data from file          S000410
c
c   common/jcfg/  icomb,iscrn,iprnt          S000420
c   common/totl/  wtotl          S000430
c   common/ulab/  ulabl          S000440
c   common/stkn/  nstat          S000450
c   common/stkp/  xstak(100),ystak(100),zstak          S000460
c   common/stkd/  idonr(100),nidnr          S000470
c   common/regn/  xcent,ycent,rinhb,xmini,xmaxi,ymini,ymaxi          S000480
c   common/stfn/  nstat          S000490
c   common/sttb/  lblok,nblk          S000500
c   common/sttp/  xstat(120),ystat(120),zstat          S000510
c   common/sttv/  icoal(3600),pstat(3600),wstat(3600)          S000520
c   common/mxmn/  xmin,xmax,ymin,ymax          S000530
c
c   character*80 ulabl          S000540
c
c   log call          S000550
c   write(8,3010)          S000560
c
c   open(7,file='bwaco.dat',form='unformatted',status='old',err=110)          S000570
c   read(7,end=130,err=120) ulabl          S000580
c   read(7,end=130,err=120) nstat,lblok,nblk,nstat,nidnr          S000590
c   read(7,end=130,err=120) wtotl,xmin,xmax,ymin,ymax          S000600
c   read(7,end=130,err=120) xcent,ycent,rinhb          S000610
c   read(7,end=130,err=120) (xstak(i),i=1,nstat)          S000620
c   read(7,end=130,err=120) (ystak(i),i=1,nstat)          S000630
c   read(7,end=130,err=120) (idonr(i),i=1,nstat)          S000640
c   read(7,end=130,err=120) (xstat(i),i=1,lblok)          S000650
c   read(7,end=130,err=120) (ystat(i),i=1,nblk)          S000660
c   read(7,end=130,err=120) (pstat(i),i=1,nstat)          S000670
c   read(7,end=130,err=120) (wstat(i),i=1,nstat)          S000680
c   read(7,end=130,err=120) (icoal(i),i=1,nstat)          S000690
c   close(7)          S000700
c
c   return          S000710
c
c   errors detected          S000720
c   110 stop, BWACO: ERROR - Unable to open bwaco.dat!          S000730
c   120 stop, BWACO: ERROR - Unable to read bwaco.dat!          S000740
c   130 stop, BWACO: ERROR - Premature end of bwaco.dat!          S000750
c
c   format statement          S000760
c   3010 format(/' geto called')          S000770
c
c   end          S000780
c-----          S000790

```

Figure A-25. Subroutines savn and geto from the interface module.

Subroutine *savn*, shown in Figure A-25, saves the results of a computation to a file.

Subroutine *geto*, shown in Figure A-25, gets the results of a previously run computation from a file.

Subroutine *pmen*, shown in Figure A-26, processes requests from the plotting menu when a computation is complete or when the user requests plotting from the main menu.

Subroutine *cnfg*, shown in Figure A-27, obtains the default configuration parameters from the file *bwaco.cfg* or calls *rcfg* if *bwaco.cfg* does not exist.

Subroutine *cnst*, shown in Figure A-28, sets the required constants.

Subroutine *mmen*, shown in Figure A-26, processes requests from the main menu.

Program *bwac*, shown in Figure A-27, is the main program. It configures the session with calls to *cnfg* and *cnst*, and initiates the main menu.

```

c
c subroutine pmen
c solicits and processes a user request from the plotting menu
c local variables:
c     irqst = user request index
c     iptyp = index of requested plot type
c     idevi = index of requested device
c     log call
c     write(8,3010)
c     restore original units
c     call rest
c     solicit user request
c 110 call spmn(irqst,iptyp,idevi)
c     if(irqst.gt.0.and.irqst.lt.5) then
c         process plotting request
c         call pits(iptyp,idevi)
c         go to 110
c     endif
c     if(irqst.eq.5) then
c         process reconfiguration request
c         call rcfg(0,3)
c         call savn
c         go to 110
c     endif
c     return
c 3010 format statement
c
c end
c
c subroutine mmrn
c solicits and processes a user request from the main menu
c local variable:
c     irqst = user request index
c     log call
c     write(8,3010)
c     solicit user request
c     iscur=0
c 110 call smmn(iscur,irqst)
c     if(irqst.eq.1) then
c         process a request to run a problem
c         call exec
c         call savn
c         call pmen
c         iscur=1
c     endif
c     if(irqst.eq.2) then
c         process a request to plot a problem
c         call geto
c         call pmen
c         iscur=1
c     endif
c     if(irqst.eq.3) then
c         process a request for session reconfiguration
c         call rcfg(0,2)
c     endif
c     if(irqst.ne.0) go to 110
c     return
c 3010 format statement
c
c end

```

Figure A-26. Subroutines pmen and mmrn from the interface module.

```

c
c      program bwac
c      is the BWACO main program
c      report the BWACO version
c      write(6,2010)
c
c      open the session log
c      open(8,file='bwaco.log',form='formatted',err=110)
c      write(8,3010)
c
c      configure the session
c      call cnfg
c
c      define the constants
c      call cnst
c
c      run the main menu
c      call mmnu
c
c      close the session log
c      write(8,3020)
c      close(8)
c
c      terminate execution
c      stop 'BWACO: Session Terminated.'
c
c      error detected
c 110 stop 'BWACO: ERROR - Unable to open bwaco.log!'
c
c      format statement
2010 format('BWACO: May 1994')
3010 format('BWACO Session Log')
3020 format('termination requested')
c
c      end
c
c
c      subroutine cnfg
c      configures the session
c
c      configuration variables;
c      icomb = pressure combination algorithm switch
c      iscrn = default display switch
c      iprnt = default printer switch
c
c      common/icfg/ icomb,iscrn,iprnt
c
c      log call
c      write(8,3010)
c
c      read data from the configuration file
c      open(7,file="bwaco.cfg",form="formatted",status="old",err=110)
c      read(7,1010,end=110,err=110) icomb,iscrn,iprnt
c      close(7)
c      go to 120
c
c      generate default configuration file
110 icomb=2
      iscrn=10
      iprnt=28
      call rcfg(1,1)
c
120 return
c
c      format statements
1010 format(i1,2i2)
3010 format(' cnfg called')
c
c      end
c
c
c      subroutine exec
c      sets up and executes the coalescence computation
c
c      log call
c      write(8,3010)
c
c      read problem input
c      call inpt
c
c      define the stacks
c      call stak
c
c      define the stations
c      call stns
c
c      make the coalescence computation
c      call copt
c
c      return
c
c      format statement
3010 format(' exec called')
c
c      end
c

```

Figure A-27. Program *bwac* and subroutines *cnfg* and *exec* from the interface module.

```

c
c      subroutine cnst
c
c      defines the constants
c
c      common/conf/  thrd,tpi
c      common/conv/  okt,ocm,odsc,ocmkt3
c      common/coni/  gamma,gmone,gpone
c      common/cona/  p1,c1,r1,t1
c      common/conp/  a0p,b0p2,c0p,alp,blp,r0p
c      common/cons/  r0s,c0s,d0s
c      common/conk/  ak0,ak(8)
c
c      log call
c      write(8,3010)
c
c      fixed constants
c      thrd=1./3.
c      tpi=8.*atan(1.)
c      twthrd=2.***thrd
c
c      conversion factor constants
c      okt=2.e6
c      ocm=0.01
c      odsc=68947.6
c      ocmkt3=2.604004e-4*twthrd
c
c      ideal gas constants
c      gamma=1.404574
c      gmone=gamma-1.
c      gpone=gamma+1.
c
c      sea-level atmospheric constants
c      call at62(0., p1,c1,r1,t1)
c
c      positive phase duration constants
c      (time in s, distance in cm, weight in kt)
c      a0p = 4.313976E-05
c      b0p2=6.261821E+08
c      c0p = 3.350000E-01
c      alp = 5.361303E+12
c      blp = 3.507575E+00
c      r0p = 4.490000E+03
c
c      shock arrival time constants
c      (time in s, distance in cm, weight in kt)
c      r0s=1.6195e4
c      c0s=3.3897e4
c      d0s=1.5000e4
c
c      kingery overpressure fit constants
c      (pressure in lb/in**2, time in s, distance in ft, weight in lb)
c      ak0 = 7.0452041
c      ak(1)=-1.6277561
c      ak(2)=-0.27399088
c      ak(3)=-6.5973136e-2
c      ak(4)= 6.5412563e-3
c      ak(5)= 4.8236359e-2
c      ak(6)=-2.0072553e-2
c      ak(7)= 3.0190449e-3
c      ak(8)=-1.5984026e-4
c
c      return
c
c      format statement
c      3010 format('' cnst called'')
c
c      end
c
c      0000010
c      0000020
c      0000030
c      0000040
c      0000050
c      0000060
c      0000070
c      0000080
c      0000090
c      0000100
c      0000110
c      0000120
c      0000130
c      0000140
c      0000150
c      0000160
c      0000170
c      0000180
c      0000190
c      0000200
c      0000210
c      0000220
c      0000230
c      0000240
c      0000250
c      0000260
c      0000270
c      0000280
c      0000290
c      0000300
c      0000310
c      0000320
c      0000330
c      0000340
c      0000350
c      0000360
c      0000370
c      0000380
c      0000390
c      0000400
c      0000410
c      0000420
c      0000430
c      0000440
c      0000450
c      0000460
c      0000470
c      0000480
c      0000490
c      0000500
c      0000510
c      0000520
c      0000530
c      0000540
c      0000550
c      0000560
c      0000570
c      0000580
c      0000590
c      0000600
c      0000610
c      0000620
c      0000630
c      0000640
c      0000650
c      0000660
c      0000670
c      0000680
c      0000690
c      0000700

```

Figure A-28. Subroutine *cnst* from the interface module.

INTENTIONALLY LEFT BLANK.

APPENDIX B:
BWACO USER'S MANUAL

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	88
BACKGROUND	88
HARDWARE AND SOFTWARE REQUIREMENTS	89
INSTALLATION	90
BWACO TUTORIAL	90
Getting Started	90
Creating Your Own Input Files	92
Obtaining Plots from Earlier Computations	95
BWACO REFERENCE	97
Coalescence Map	97
Contour Levels	97
Files and Program Execution	97
Inhabited Building Limit	98
Initial Donors	98
Input Files	98
Menus	99
Peak Overpressure Map	101
Plotting Distortion	101
Printers	101
Region of Interest	101
Stations	102
Template	102
Video Displays	102

INTRODUCTION

This manual provides background information on the BWACO methodology, outlines hardware and software requirements and directs installation of the code. The BWACO Tutorial gives instructions for running sample problems and creating and running your own problems. In addition, the BWACO Reference describes all BWACO functions and features in alphabetical order. Finally, a detailed description of BWACO, listing most of the source code, is provided. Users who experience difficulty should contact the authors at (410) 278-6214/6241.

BACKGROUND

The BWACO program was written to predict regions of blast wave coalescence associated with the sequential mass detonation of distributed ammunition stacks positioned as specified by the user.

It accomplishes this by applying the assumption that communication of mass detonation between stacks is controlled by a user-supplied propagation velocity (which might represent the velocity of fragments from the donor stack). BWACO uses this propagation velocity to determine the order and timing of detonation of the stacks. It recognizes that the identity of the initial donor is not generally known and, unless a single initial donor has been specified by the user, considers all possible initial donors in order to predict the worst-case environment at any point.

Then, under the assumption that blast waves propagate independently of one another, BWACO applies the criterion that coalescence occurs wherever and whenever the shock associated with one blast wave encroaches into the positive overpressure phase of another blast wave. This assumption generally causes premature identification of coalescence. When coalescence is detected at a point, it determines the combined peak overpressure by first combining, at their center of charge, the stacks producing the waves which coalesce and then computing the peak overpressure associated with the combined stacks at that point. This method is suitable, at least, for closely spaced stacks.

BWACO graphically reports regions of coalescence, showing the effective total explosive weight and the peak overpressure throughout the region of interest.

In applying BWACO, it is important to remain aware of limitations imposed by the assumptions. The accuracy of BWACO has not been assessed except in the case of closely spaced charges. As the separation between stacks is increased, the accuracy, especially that of the peak overpressure predictions, is expected to decrease.

Detailed background information can be found in the references.

HARDWARE AND SOFTWARE REQUIREMENTS

BWACO should run in the DOS environment on any personal computer with an 80?86 microprocessor. However, to minimize the running times for large problems, a high-speed computer with at least an 80486 microprocessor is recommended.

An editor (such as the DOS editor) that produces pure ASCII text (without hidden characters) is required to create problem input files. A word processing program should not be used unless the file can be saved in ASCII format.

BWACO's graphics supports a number of different video displays and printers. The displays are listed in the following table:

Type	Mode	Resolution	Colors
CGA	4	320x200	4
CGA	5	320x200	4
CGA	6	640x200	BW
Hercules		720x348	BW
EGA	13	320x200	16
EGA	14	640x200	16
EGA	15	640x350	BW
EGA	16	640x350	16
VGA	17	640x480	BW
VGA	18	640x480	16
ATI	81	640x480	16
ATI	82	752x410	16
ATI	83	800x560	16

The following printers are supported:

- single-density printer
- double-density printer
- quad-density printer
- Epson printer (plot mode)
- Postscript printer
- Hewlett-Packard Laser Jet
- Hewlett-Packard Think Jet

Hard copies requested during a BWACO session are automatically sent to the printer at the end of the session.

INSTALLATION

Please follow these installation instructions very carefully. Before installing BWACO, create a directory to hold the BWACO files. For example, you can create the directory **bwaco** by entering the following line:

```
>mkdir bwaco
```

Make that directory the current directory.

```
>cd bwaco
```

Insert the BWACO installation diskette into the appropriate drive. If you inserted it into drive **a:**, enter the following line:

```
>a:install a
```

If you inserted it into drive **b:**, enter the following line:

```
>b:install b
```

This will cause four files to be copied to your new directory, after which the installation is complete.

BWACO TUTORIAL

Getting Started

BWACO always looks for input in a file with the extension **inp**. A sample input file has been supplied and running this problem before trying your own is recommended. The file **three.inp** describes three 25,000-lb stacks of ammunition, separated by 200 meters, lying at the vertices of an equilateral triangle. To start a BWACO session and run this problem, simply enter the following line:

```
>bc three
```

The first time you run BWACO the Configuration Menu will appear. This menu, which is similar to the other BWACO menus, requires you to enter a single-digit number to select one of the menu options:

BWACO CONFIGURATION MENU

Default Display: VGA Mode 18 640x480/16
Default Printer: Postscript Printer

Enter 1 to change the default display
2 to change the default printer
0 to proceed to the Main Menu

The Configuration Menu allows you to tell BWACO which display and printer are installed on your system. The default display is **VGA Mode 18** and the default printer is **Postscript Printer**. If your display or printer is different, make changes now. If you do not specify a display or printer, BWACO will save and use these defaults. To change the display or printer, simply enter either 1 or 2 and make your selection from the list displayed. BWACO will remember your specifications in all subsequent sessions. To leave the menu, simply strike the **enter** key. Typing 0 is not necessary. After you have completed the initial configuration, the Main Menu will appear:

BWACO MAIN MENU

Enter 1 to run the current problem
2 to reconfigure the session
0 to end the session

To run the problem configuration described in **three.inp**, enter 1. You will have to wait some period of time while the problem runs. BWACO will keep you informed of what it is doing. When the computation is complete, BWACO will tell you how long it took to run on your system and the Plotting Menu will appear:

BWACO PLOTTING MENU

Enter 1 to plot the Coalescence Map (Screen)
2 to plot the Coalescence Map (Printer)
3 to plot the Peak Overpressure Map (Screen)
4 to plot the Peak Overpressure Map (Printer)
5 to reconfigure the session
0 to return to the Main Menu

You can read about the Coalescence and Peak Overpressure Maps in the reference section of this manual. Produce the Coalescence Map on your screen by entering 1. Strike any key to return to the Plotting Menu. Enter 2 to obtain a hard copy. All hard copies requested during a session will be printed when the session is terminated. Try the Peak Overpressure Map in a similar fashion. If you experience problems with plotting, you may have made the wrong configuration selections. You can correct them by entering 5 from the Plotting Menu.

When you are finished plotting, return to the Main Menu:

BWACO MAIN MENU

Enter 1 to run the current problem
2 to plot the current problem
3 to reconfigure the session
0 to end the session

This menu now provides the additional option of plotting the results of the problem without rerunning. End the session by striking the **enter** key. You may receive a query regarding your printer. If so, simply strike **enter** again. Any hard copies you requested will print at this time.

Creating Your Own Input Files

BWACO requires its input file to have a very specific format. Before creating a new input file, let's look at the format of the file **three.inp**.

```
Three 25-klb Stacks, 200-m Separation (three.inp)
|---Xmin---|---Xmax---|---Ymin---|---Ymax---|
      -700.      700.      -700.      700.
|---Xs---|---Ys---|---Ws---|---Vs---| D
      100.    -57.74    25000.    2500.0    1
     -100.    -57.74    25000.    2500.0    1
       0.     115.48    25000.    2500.0    1
```

The first line of this file contains a descriptive problem title of the user's choice. It may contain up to 80 characters. It is used to label the plots. As a convenience, we have included the input file name in parentheses at the end of the title. The second line is a ruler line delimiting the fields in which the bounds of the region of interest are placed. These bounds, which appear on the third line, determine the region in which the stations are distributed. The fourth line is another ruler line specifying the fields in which stack data is placed. The three stacks are described, one to a line, on the remaining lines. The stack coordinates, the TNT equivalent explosive weight, and the propagation velocity are specified. In addition, the D (initial donor) switch is set to 1 to ensure that each stack may act as an initial donor.

Let's see how problem input is generated. The net explosive weight for each stack should be expressed in equivalent pounds of TNT, distances in meters and propagation velocities in meters per second. Suppose we have four 25,000-lb ammunition stacks arranged as shown in Figure B-1. First, we must define a Cartesian coordinate system. The orientation and origin location of our coordinate system is arbitrary and may be chosen to suit our convenience. One possibility, along with the coordinates of each stack, is shown in Figure B-2. In addition to this, we also have to know the propagation velocity

associated with each stack. Since no better guidance is presently available, we have fixed this at 2,500 m/s.

We now know enough to generate the input file. In order to create your own problem input files, you will need to use an editor that produces pure ASCII files (without hidden characters), such as the DOS editor. A word processing program should not be used. As a convenience, an input template has been included with BWACO. To start out, simply copy this file to the new input file, as follows:

```
>copy template.inp four.inp
```

Then, edit the file **four.inp**. Initially, it looks like this:

```
|---Xmin---| |---Xmax---| |---Ymin---| |---Ymax---|  
|---Xs---| |---Ys---| |---Ws---| |---Vs---| D
```

First we'll add a descriptive title:

```
Four 25-klb Stacks, 200-m Separation (four.inp)  
|---Xmin---| |---Xmax---| |---Ymin---| |---Ymax---|  
|---Xs---| |---Ys---| |---Ws---| |---Vs---| D
```

Next, since we don't know how big the region of interest should be, we'll let BWACO determine its limits by specifying X and Y bounds which are equal:

```
Four 25-klb Stacks, 200-m Separation (four.inp)  
|---Xmin---| |---Xmax---| |---Ymin---| |---Ymax---|  
0. 0. 0. 0.  
|---Xs---| |---Ys---| |---Ws---| |---Vs---| D
```

Finally, we'll specify the stack coordinates, weights and propagation velocities, setting the initial donor switch to 1 for each stack:

```
Four 25-klb Stacks, 200-m Separation (four.inp)  
|---Xmin---| |---Xmax---| |---Ymin---| |---Ymax---|  
0. 0. 0. 0.  
|---Xs---| |---Ys---| |---Ws---| |---Vs---| D  
-100. -100. 25000. 2500. 1  
100. -100. 25000. 2500. 1  
-100. 100. 25000. 2500. 1  
100. 100. 25000. 2500. 1
```

This completes the generation of the input file **four.inp**. Try running this problem:

```
>bc four
```

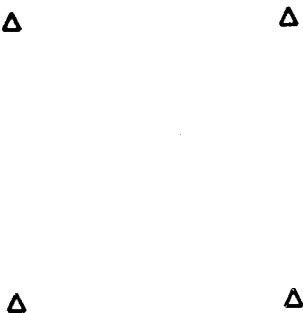


Figure B-1. Arrangement of four 25,000-lb ammunition stacks.

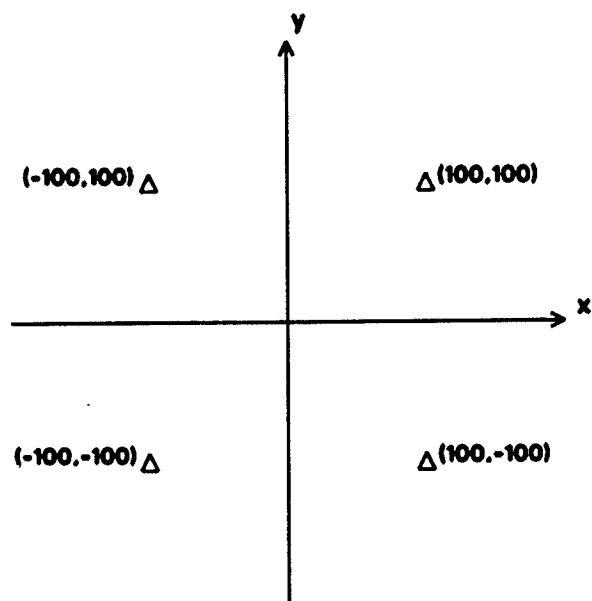


Figure B-2. A coordinate system for this arrangement.

The plots produced will be somewhat distorted. This occurs because the algorithm used to extend the axes to a natural termination point treats the X and Y axes differently. To eliminate the distortion, it is necessary to rerun the problem after modifying the input file to specify region limits:

```
Four 25-klb Stacks, 200-m Separation (four.inp)
|---Xmin---||---Xmax---||---Ymin---||---Ymax---|
    -800.        800.        -800.        800.
|---Xs---||---Ys---||---Ws---||---Vs---|      D
    -100.        -100.        25000.      2500.      1
    100.        -100.        25000.      2500.      1
    -100.        100.        25000.      2500.      1
    100.        100.        25000.      2500.      1
```

With this modification (which is not always necessary when allowing BWACO to determine the extent of the region of interest) undistorted plots, as shown in Figures B-3 and B-4, are produced.

Obtaining Plots From Earlier Computations

We can easily obtain new plots from the problem described in `three.inp` run earlier. Be sure to wait until all previously requested plots have printed, otherwise the files may be deleted before they print. Simply enter the following line:

```
>bc three
```

The Main Menu appears as follows:

```
BWACO MAIN MENU
```

```
Enter 1 to run the current problem
      2 to plot the previous problem
      3 to reconfigure the session
      0 to end the session
```

The previously obtained results may be plotted by entering 2 and making the desired selections from the Plotting Menu. End the session by striking `enter` twice.

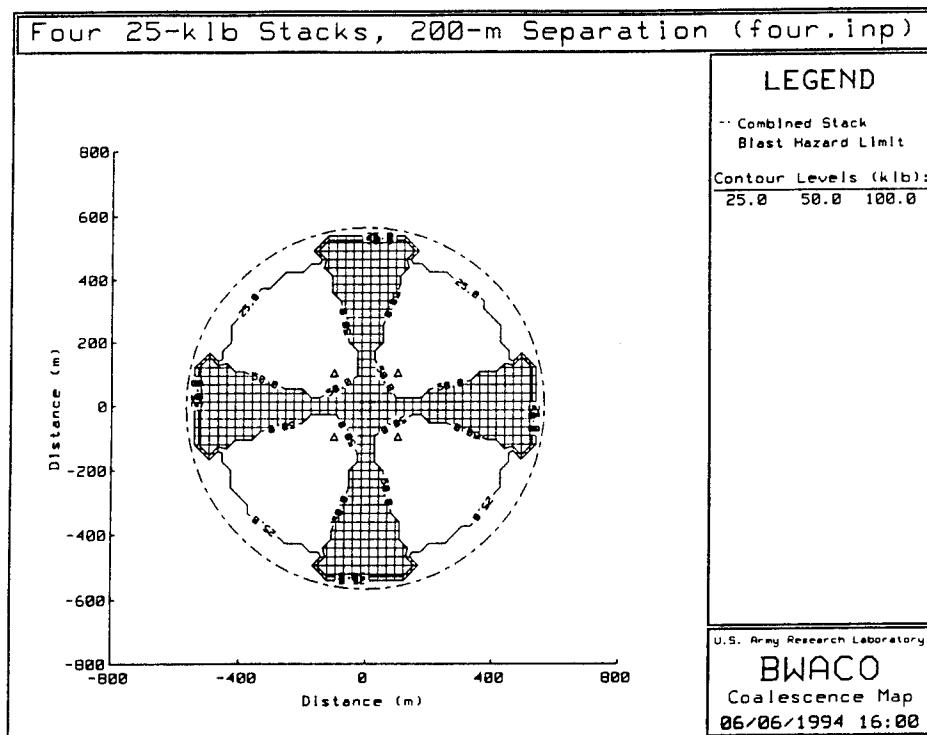


Figure B-3. Coalescence Map.

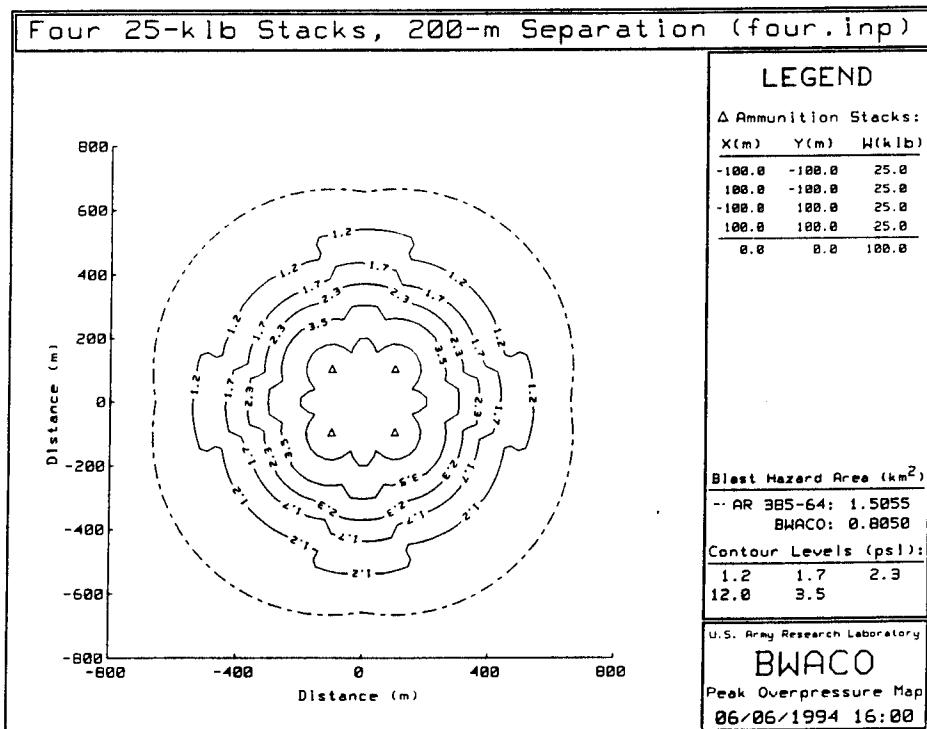


Figure B-4. Peak Overpressure Map.

BWACO REFERENCE

Coalescence Map

An example of a Coalescence Map is shown in Figure B-3. Regions in which coalescence is detected are hatched and explosive weight contours are drawn. Each contour is labeled to indicate the total explosive weight contributing to the worst-case blast environment within that contour. The outermost contour follows the 0.9 or 1.2 psi overpressure limit for inhabited buildings defined in Army Regulation (AR) 385-64. For comparison purposes, a circle, centered at the center of charge of all the stacks and having the inhabited building distance for the combined explosive weight of all the stacks as its radius, is also plotted. As many contour levels as space permits are listed in the legend. (See **Peak Overpressure Map** and **Plotting Distortion**.)

Contour Levels

The contour levels used in the Coalescence Map are listed in the legend (space permitting). These levels describe the total explosive weight contributing to the worst-case blast environment within the contour. The contour levels used in the Peak Overpressure Map are those specified in AR 385-64 for the protection of a variety of assets.

Files and Program Execution

The file **bc.bat** is the DOS batch file that executes BWACO and manages its files. The file **bwaco.exe** is the executable BWACO program. The files ***.inp** are the user's problem specification input files (where * represents the user's descriptive file name). (See **Input Files**.) The files ***.dat** contain the plotting data, enabling BWACO to plot results without rerunning problems. The files ***.log** contain logs of BWACO's most recent session pertaining to the problem identified by *. The user starts a BWACO session by entering **bc ***. The batch file first deletes any old plotting files. For this reason, a BWACO session should never be started until all plots from the previous session have printed. Next, it copies ***.inp** and ***.dat** (if it exists) to **bwaco.inp** and **bwaco.dat**, respectively. It then executes **bwaco.exe**, which creates the files **bwaco.dat** and **bwaco.log**. On termination of the session, the batch file deletes **bwaco.inp**, moves **bwaco.dat** and **bwaco.log** to ***.dat** and ***.log**, respectively and sends all plotting files to the printer using the **print** command.

Inhabited Building Limit

AR 385-64 defines permissible overpressure exposure levels for a variety of assets. The lowest such levels are those commonly associated with inhabited buildings (0.9 to 1.2 psi). The levels are also given as scaled distances using classical blast scaling laws. The minimum distance is further limited by consideration of hazardous fragment density. BWACO combines all the stacks by summing their explosive weights and determining their center of charge. Using the combined weight, it computes the inhabited building distance. Using the center of charge, it plots the combined inhabited building circle on the Coalescence Map. On the Peak Overpressure Map, it plots a contour produced by centering inhabited building distance circles at each stack. This represents the method currently required by the regulation and allows comparison with BWACO's predictions.

Initial Donors

When detonation propagates from one stack to another, the first stack of such a pair is referred to as the donor. The first of all the stacks to detonate is called the initial donor. BWACO allows the user to specify which stacks may act as initial donors. It is not generally possible to predict which ammunition stack will act as an initial donor when a mass-detonation accident occurs. Therefore, in most applications, the user should specify that all stacks may act as initial donors. BWACO will issue a warning whenever any stacks may not act as initial donors. (See **Input Files**.)

Input Files

An editor that produces pure ASCII files (without hidden characters), such as the DOS editor, must be used to create problem input files. A word processing program should not be used unless the file can be saved in ASCII format. BWACO looks for input in a file with the extension **inp**. A descriptive file name (up to eight characters) may be used. The file must conform to the format (see **Template**) shown in the following example:

```
Four 25-klb Stacks, Separation=200 m (four.inp)
|---Xmin---|---Xmax---|---Ymin---|---Ymax---|
      0.          0.          0.          0.
|---Xs---|---Ys---|---Ws---|---Vs---|   D
      -100.      -100.      25000.      2500.      1
      100.       -100.      25000.      2500.      1
     -100.       100.       25000.      2500.      1
      100.       100.       25000.      2500.      1
```

The first line of the file contains a descriptive problem title of the user's choice. It may contain up to 80 characters. It is used to label the plots, centered in the plot label box at the top of both the Coalescence and Peak Overpressure Maps. The second line is a ruler line delimiting the fields in which the bounds of the region of interest are placed. These bounds, which appear on the third line, determine the region in which the stations are distributed. When the input bounds for either coordinate axis are equal (Xmin=Xmax or Ymin=Ymax), BWACO computes appropriate bounds. The fourth line is another ruler line specifying the fields in which stack data is placed. Up to 100 stacks are described, one to a line, on the remaining lines. The stack coordinates, net equivalent explosive weight, and propagation velocity are specified. In addition, the D (initial donor) switch is set to 0 or 1 to specify whether or not each stack may act as an initial donor. For most problems, the initial donor switch should be set to 1 for all stacks. (See **Initial Donors**.)

Menus

Menu selections are made by entering a single-digit number at the prompt. A menu is exited by entering 0 or its equivalent, simply striking **enter**.

The Configuration Menu allows selection of the appropriate video display and printer. It may appear in any of the following three forms:

BWACO CONFIGURATION MENU

Default Display: VGA Mode 18 640x480/16
Default Printer: Postscript Printer

Enter 1 to change the default display
2 to change the default printer
0 to proceed to the Main Menu

BWACO CONFIGURATION MENU

Default Display: VGA Mode 18 640x480/16
Default Printer: Postscript Printer

Enter 1 to change the default display
2 to change the default printer
0 to return to the Main Menu

BWACO CONFIGURATION MENU

Default Display: VGA Mode 18 640x480/16
Default Printer: Postscript Printer

Enter 1 to change the default display
2 to change the default printer
0 to return to the Plotting Menu

The Main Menu allows selection of the problem execution, plotting or reconfiguration functions, as available. It may appear in any of the following three forms:

BWACO MAIN MENU

Enter 1 to run the current problem
2 to reconfigure the session
0 to end the session

BWACO MAIN MENU

Enter 1 to run the current problem
2 to plot the current problem
3 to reconfigure the session
0 to end the session

BWACO MAIN MENU

Enter 1 to run the current problem
2 to plot the previous problem
3 to reconfigure the session
0 to end the session

The Plotting Menu allows selection of Coalescence or Peak Overpressure Maps on either the screen or the printer. It appears as follows:

BWACO PLOTTING MENU

Enter 1 to plot the Coalescence Map (Screen)
2 to plot the Coalescence Map (Printer)
3 to plot the Peak Overpressure Map (Screen)
4 to plot the Peak Overpressure Map (Printer)
5 to reconfigure the session
0 to return to the Main Menu

Peak Overpressure Map

An example of a Peak Overpressure Map is shown in Figure B-4. Peak overpressure contours are labeled to indicate the worst-case values at each point within the region of interest. The contour levels, which are listed in the legend, are those specified in AR 385-64 for the protection of a variety of assets. For comparison purposes, an additional contour representing the application of current regulatory requirements is also shown. The positions and equivalent explosive weights of the stacks as well as the center of charge and total equivalent explosive weight are also shown in the legend. (See **Coalescence Map** and **Plotting Distortion**.)

Plotting Distortion

Plotting distortion occurs when the algorithm used to extend the axes to a natural termination point treats the X and Y axes differently. To eliminate the distortion, it is necessary to rerun the problem after modifying the input file to specify different region limits. (See **Coalescence Map** and **Peak Overpressure Map**.)

Printers

The following printers are supported:

```
single-density printer
double-density printer
quad-density printer
Epson printer (plot mode)
Postscript printer
Hewlett-Packard Laser Jet
Hewlett-Packard Think Jet
```

Region of Interest

The region of interest is the region covered by the Coalescence and Peak Overpressure Maps. It is specified by giving values of Xmin, Xmax, Ymin, and Ymax in the user's input file. If the user specifies values of Xmin and Xmax which are equal, BWACO will compute the range of the region of interest in the X direction. An analogous algorithm applies in the Y direction. (See **Stations**.)

Stations

When running a problem, BWACO positions 3,600 stations within the region of interest. These are not generally apparent to the user. Coalescence is checked at each of these stations for each initial donor. (See **Region of Interest**.)

Template

The file **template.inp** is a template for preparing input files. It can be copied as the first step in preparing a new input file. It includes the two ruler lines and appears as follows:

```
|---Xmin---| |---Xmax---| |---Ymin---| |---Ymax---|  
|---Xs---| |---Ys---| |---Ws---| |---Vs---|      D
```

(See **Input Files**.)

Video Displays

The supported displays are listed in the following table:

Type	Mode	Resolution	Colors
CGA	4	320x200	4
CGA	5	320x200	4
CGA	6	640x200	BW
Hercules		720x348	BW
EGA	13	320x200	16
EGA	14	640x200	16
EGA	15	640x350	BW
EGA	16	640x350	16
VGA	17	640x480	BW
VGA	18	640x480	16
ATI	81	640x480	16
ATI	82	752x410	16
ATI	83	800x560	16

<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>
2	ADMINISTRATOR DTIC ATTN DTIC DDA CAMERON STATION ALEXANDRIA VA 22304-6145	1	DIR USA TRADOC ANALYSIS CMD ATTN ATRC WSR WSMR NM 88002-5502
1	CDR USAMC ATTN AMCAM 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001	1	CMDT US ARMY INFANTRY SCHOOL ATTN ATSH CD SECURITY MGR FORT BENNING GA 31905-5660
1	DIR USARL ATTN AMSRL OP SD TA RECORDS MGMT 2800 POWDER MILL RD ADELPHI MD 20783-1145		<u>ABERDEEN PROVING GROUND</u>
3	DIR USARL ATTN AMSRL OP SD TL TECH LIB 2800 POWDER MILL RD ADELPHI MD 20783-1145	2	DIR USAMSAA ATTN AMXSY D AMXSY MP H COHEN
1	DIR USARL ATTN AMSRL OP SD TP TECH PUBS BR 2800 POWDER MILL RD ADELPHI MD 20783-1145	1	CDR USATECOM ATTN AMSTE TC
2	CDR US ARMY ARDEC ATTN SMCAR TDC PICATINNY ARSNSL NJ 07806-5000	1	DIR USAERDEC ATTN SCBRD RT
1	DIR BENET LABS ATTN SMCAR CCB TL WATERVLIET NY 12189-4050	1	CDR USACBDCOM ATTN AMSCB CII
1	DIR USA ADVANCED SYSTEMS R&A OFC ATTN AMSAT R NR MS 219 1 AMES RESEARCH CENTER MOFFETT FIELD CA 94035-1000	1	DIR USARL ATTN AMSRL SL I
1	DIR US ARMY MICOM ATTN AMSMI RD CS R DOC REDSTONE ARSNSL AL 35898-5010	5	DIR USARL ATTN AMSRL OP AP L
1	CDR US ARMY TACOM ATTN AMSTA JSK ARMOR ENG BR WARREN MI 48397-5000		

<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>
1	DIRECTOR US ARMY TECHNICAL CENTER FOR EXPLOSIVE SAFETY ATTN: SMCAC-ESL/C DOYLE SAVANNA IL 61074-9639	3	SOUTHWEST RESEARCH INSTITUTE ATTN: P BOWLES C OSWALD E ESPARZA 6220 CULEBRA ROAD PO DRAWER 28510 SAN ANTONIO TX 78228-0510
3	US ARMY WATERWAYS EXPERIMENT STATION ATTN: CEWES-SE/ K DAVIS M FORD C JOACHIM 3909 HALLS FERRY RD VICKSBURG MS 39180-6199	1	DIRECTOR EXPLOSIVES HAZARD REDUCTION DIRECTORATE ATTN: J JENUS ASC/YOCO (EHR) EGLIN AFB FL 32542
3	CHAIRMAN DEPARTMENT OF DEFENSE EXPLOSIVE SAFETY BOARD ATTN: DDESB-KT/ J WARD (2 CP) C CANADA 2461 EISENHOWER AVE ALEXANDRIA VA 22331-0600	2	US ARMY CORPS OF ENGINEERS ATTN: CEHND-ED-CS/ R WRIGHT R HASSE PO BOX 1600 HUNTSVILLE AL 35807-4301
1	DEFENSE AMMUNITION LOGISTICS ACTIVITY ATTN: AMCPM-AL/R ROSSI PICATINNY ARSENAL NJ 07806-5000	3	COMMANDER DEFENSE NUCLEAR AGENCY ATTN: T KENNEDY D LINGER G ULLRICH WASHINGTON DC 20305-1000
1	OFFICER IN CHARGE WHITE OAK LABORATORY NAVAL SURFACE WARFARE CENTER ATTN: R15/M SWISDAK 10901 NEW HAMPSHIRE AVE SILVER SPRING MD 20902-5000	1	COMMANDER FIELD COMMAND DNA ATTN: FCTTS (MCCRORY) KIRTLAND AFB NM 87117-5669
2	NAVAL FACILITIES ENGINEERING SERVICES CENTER ATTN: CODE LS1/ J TANCRETO R MURTHE PORT HUENEME CA 93043	1	HQDA (SARD) ATTN: MR MLARCHIK WASHINGTON DC 20310-0001
2	MAXWELL S-CUBED DIVISION ATTN: C NEEDHAM S HIKIDA 2501 YALE BLVD SE SUITE 300 ALBUQUERQUE NM 87106-4215	1	COMMANDER US ARMY NUCLEAR AND CHEMICAL AGENCY ATTN: MONA-NU 7150 HELLER LOOP SUITE 101 SPRINGFIELD VA 22150-3198
		1	ABERDEEN RESEARCH CENTER ATTN: N ETHRIDGE PO BOX 548 ABERDEEN MD 21001

NO. OF
COPIES ORGANIZATION

1 CARPENTER RESEARCH CORPORATION
ATTN: J CARPENTER
PO BOX 2490
ROLLING HILLS ESTATES CA 90274

1 LOGICON R&D ASSOCIATES
ATTN: G GANONG
PO BOX 9335
ALBUQUERQUE NM 87119

1 TECH REPS INC
ATTN: F MCMULLAN
5000 MARBLE NE SUITE 222
ALBUQUERQUE NM 87110

ABERDEEN PROVING GROUND

20 DIR, USARL
ATTN: AMSRL-WT-T/W MORRISON
AMSRL-WT-TB/
R FREY
O LYMAN
T WATSON
F GREGORY
V BOYLE
W HILLSTROM
E MCDOUGAL
W LAWRENCE
K BENJAMIN
T DORSEY
AMSRL-WT-TD/
A DAS GUPTA
J SANTIAGO
AMSRL-WT-NC/
R LOTERRO
R RALEY
J SULLIVAN
S SCHRAML
AMSRL-CI-CA/
A CELMINS
C ZOLTANI

NO. OF
COPIES ORGANIZATION

1 SNPE
ATTN: L ALLAIN
BP2 91710 VERT-LE-PETIT
FRANCE

2 MINISTRY OF DEFENSE
EXPLOSIVES STORAGE AND
TRANSPORTATION COMMITTEE
ATTN: J CONNOR
M GOULD
ROOM 1818/EMPRESS STATE BUILDING
LONDON SW6 1TR
ENGLAND

1 AGENCY FOR DEFENSE DEVELOPMENT
ATTN: S - Y SONG
PO BOX 35
YUSEONG TAEJON 305-600
REPUBLIC OF KOREA

1 AUSTRALIAN ORDNANCE COUNCIL DOD
ATTN: J GOOLD
CANBERRA ACT 2600
AUSTRALIA

1 NATIONAL DEFENSE RESEARCH
ESTABLISHMENT FOA
ATTN: R FORSEN
PO BOX 551
TUMBA S-14725
SWEDEN

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number ARL-TR-645 Date of Report December 1994

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.)

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate.

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

Organization _____

CURRENT ADDRESS Name _____

Street or P.O. Box No. _____

City, State, Zip Code _____

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

Organization _____

OLD ADDRESS Name _____

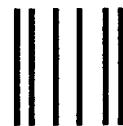
Street or P.O. Box No. _____

City, State, Zip Code _____

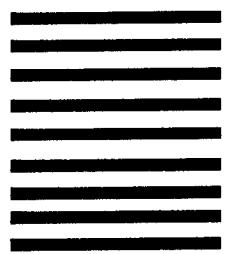
(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

A vertical stack of ten thick, black, horizontal lines, used for postal processing.

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 0001, APG, MD

Postage will be paid by addressee

Director
U.S. Army Research Laboratory
ATTN: AMSRL-OP-AP-L
Aberdeen Proving Ground, MD 21005-5066